

AD A111317

**RADC-TR-81-309**  
**Final Technical Report**  
**November 1981**



# **NATIONAL SOFTWARE WORKS TOOL INTEGRATION STUDIES**

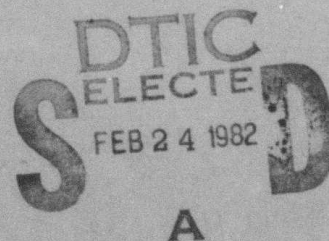
**General Systems Group, Inc.**

Sponsored by  
Defense Advanced Research Projects Agency (DoD)  
ARPA Order No. 3686

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

**ROME AIR DEVELOPMENT CENTER**  
**Air Force Systems Command**  
**Griffiss Air Force Base, New York 13441**



82 02 24 029

DTIC FILE COPY

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-309 has been reviewed and is approved for publication.

APPROVED:

*Patricia J. Baskinger*  
PATRICIA J. BASKINGER  
Project Engineer

APPROVED:

*John J. Marciniak*  
JOHN J. MARCINIAK, Colonel, USAF  
Chief, Information Sciences Division

FOR THE COMMANDER:

*John P. Huss*  
JOHN P. HUSS  
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISCP) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

NATIONAL SOFTWARE WORKS TOOL INTEGRATION STUDIES

General Systems Group, Inc.

Contractor: General Systems Group, Inc.  
Contract Number: F30602-79-C-0106  
Effective Date of Contract: 12 December 1978  
Contract Expiration Date: 15 May 1981  
Short Title of Work: National Software Works Tool  
Integration Studies  
Program Code Number: 9P10  
Period of Work Covered: Dec 78 - Apr 81  
  
Principal Investigator: Normand Rasmussen  
Phone: 603 893-1000  
  
Project Engineer: Patricia Baskinger  
Phone: 315 330-7007

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Patricia Baskinger (RADC/ISCP), Griffiss AFB NY 13441 under Contract F30602-79-C-0106.

411664

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-81-309	2. GOVT ACCESSION NO. <b>AD-A111 317</b>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NATIONAL SOFTWARE WORKS TOOL INTEGRATION STUDIES	5. TYPE OF REPORT & PERIOD COVERED Final Technical Report Dec 78 - Apr 81	6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) General Systems Group, Inc.	8. CONTRACT OR GRANT NUMBER(s) F30602-79-C-0106	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington VA 22209	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63728F 25310103	
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISCP) Griffiss AFB NY 13441	12. REPORT DATE November 1981	13. NUMBER OF PAGES 182
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same	15. SECURITY CLASS. (of this report) UNCLASSIFIED	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Patricia J. Baskinger (ISCP)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Network Operating Systems      Resource Sharing Software Systems                  Operating Systems Computer Network Resources		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This final technical report documents the following specific activities of the subject contract: (1) Review of the NSW project organizational structure, including roles, responsibilities and contractor assignments, (2) Description of the NSW development process, (3) Survey of the NSW system architecture and user functionality, (4) Establishment of the current operational status of NSW including the chronology of major events, and finally (5) Serve as an introductory NSW tutorial for new		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

users.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## TABLE OF CONTENTS

	Page
1.0 Preface . . . . .	7
2.0 Introduction . . . . .	9
2.1 History . . . . .	12
2.2 Policy . . . . .	15
3.0 NSW Project Organization . . . . .	17
3.1 Organizational Structure . . . . .	18
3.2 NSW Contractors. . . . .	20
4.0 NSW System Architecture . . . . .	22
4.1 The ARPANET . . . . .	23
4.2 NSW - A Network Operating System . . . . .	28
4.3 Host Families . . . . .	29
4.4 NSW Software Architecture . . . . .	30
4.4.1 NSW Software Components . . . . .	34
4.4.2 DMC Responsibilities . . . . .	36
4.4.3 Participation of Hosts in NSW Systems . . . . .	38
4.4.4 Levels of Host Family Participation . . . . .	40
4.4.5 The NSW User System Configuration . . . . .	42
5.0 NSW System Development . . . . .	45
5.1 NSW System Configurations . . . . .	48
5.2 NSW Testing Hierarchy . . . . .	50
5.3 NSW System Release and Assessment . . . . .	51
5.4 Quality Assurance Testing . . . . .	57
5.5 STR Processing . . . . .	61
5.6 Configuration Management . . . . .	65
5.7 NSW Release History . . . . .	70
6.0 NSW User Interface . . . . .	74
6.1 Overview . . . . .	75
6.1.1 Project Management . . . . .	76
6.1.2 NSW File System . . . . .	85
6.1.3 NSW Tools . . . . .	88
6.1.4 NSW Command Language . . . . .	95

6.2	NSW User System Status (May 1981)	96
6.2.1	NSW Command Language Status	97
6.2.2	NSW File System	98
6.2.3	NSW Tools	102
6.2.4	User Documentation	113
6.2.5	Future Directions and Problem Areas	115
6.2.6	NSW Availability	117
7.0	NSW System Operations	118
7.1	Overview of NSWOPS Responsibilities	119
7.2	Status of NSW Operations	120
8.0	AFLC Technology Demonstration	125
9.0	Notes and References	126
Appendix A:	GSG Bibliography	130
Appendix B:	Generic Configuration Items	133
Appendix C:	An Overview of the NSW Help Facility	153
Appendix D:	Sample "Weekly Usage Report"	161

List of Figures	Page
3-1 NSW Organizational Structure. . . . .	21
4-1 ARPANET Geographic Map . . . . .	26
4-2 ARPANET Logical Map . . . . .	27
4-3 Component Communication Via MSG . . . . .	32
4-4 NSW's Star-Shaped Control Structure . . . . .	33
4-5 The NSW User System Configuration . . . . .	44
5-1 Stages of NSW System Development . . . . .	47
5-2 NSW System Configurations . . . . .	49
5-3 NSW System Development Life Cycle . . . . .	56
5-4 STR Resolution: Organizational Interactions. . . . .	64
6-1 Project Node A and Tool/File Rights . . . . .	79
6-2 Node A Creates Nodes B, C and D . . . . .	80
6-3 Node D Creates Nodes DA, DB and DC . . . . .	81
6-4 Node A Deletes Node B . . . . .	82
6-5 Node A Revokes Node Creation Right from Node D . . . . .	83
6-6 Node A Assigns File Right B.A to Node DB . . . . .	84
6-7 Movement and Translation of NSW File During EXPORT . . . . .	87
6-8 Assigning a workspace to An XED Tool Instance . . . . .	90
6-9 File Movement and Translation Resulting from XED Read . . . . .	91
6-10 Creation of New File in Workspace . . . . .	92
6-11 Modification of NSW File in the workspace . . . . .	93

6-12 Delivery of XED Output Files into NSW Filespace . . . 94

## List of Tables

## Page

6-1 Host Family File Types . . . . .	100
6-2 NSW Tools by Host Family . . . . .	104
6-3 Tool Statistics by NSW Host and Tool Type . . . . .	105
6-4 Alphabetic Listing of NSW Tools by Host . . . . .	106
6-5 NSW Tools by Generic Tool Type and Host . . . . .	109

## 1.0 Preface

The General Systems Group (GSG) has been involved in the National Software Works (NSW) project since 1976. During the period 1976 - 1981, GSG has been awarded four (4) NSW contracts. GSG's participation in the NSW project and responsibilities, vis-a-vis assessment and operation of the NSW system, have grown appreciably during each successive contract period.

GSG has, in the past, advocated and successfully demonstrated the viability of an independent testing and operations organizations for the NSW system. Based on these successes, GSG was asked to recommend a plan for managing the development, operation and maintenance of the NSW software system [1]. GSG's recommendation became an integral part of the "NSW Management Plan" [2], prepared and distributed by the sponsor organizations (RADC and ARPA). Implementation of this plan has been a major goal during the present contract period covered by this report. This report concentrates heavily on the role GSG has played in implementing and evolving the system development approach called out by the "NSW Management Plan". GSG is contractually responsible for fulfilling the following two (2) roles:

- Product Development Contractor (PDC)
- NSW System Operations (NSWOPS)

These organizations, staffed by GSG personnel, are respectively responsible for two of the four stages of the NSW system development process (see section 5.0), namely:

- Productization and Quality Assurance (PDC), and
- Production operation of the NSW system (NSWOPS).

This report attempts to meet a number of goals simultaneously. Discussing GSG's NSW activities and documenting our project contributions during the contract period are one goal. However, this report is also intended to serve the following purposes as well:

- Review the NSW project organizational structure, including roles, responsibilities and contractor assignments
- Describe the NSW system development process

- Survey NSW's system architecture and user functionality
- Establish the current operational status of NSW;  
indicate the major chronology of events.
- Serve as an introductory NSW tutorial for new users.

## 2.0 Introduction

The National Software Works (NSW) is a pioneering R & D project, jointly sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force. The NSW system is a network operating system which attempts to achieve integrated resource sharing in a heterogeneous computer networking environment.

A computer network is a collection of autonomous computers, called "hosts", interconnected by a common communications subsystem. Hosts may be geographically dispersed (i.e., tens, hundreds or even thousands of miles apart), and of widely different make and/or manufacture. Computers of different manufacture are usually software incompatible. That is, programs which run on one computer will not be readily portable to another for execution. Also, the syntax and interpretation of stored data (programs and the data on which they operate) vary considerably. Such incompatibilities manifest themselves as differences in:

- Hardware architecture and/or
- Operating system, including
  - Program execution environment
  - File system, and
  - Command language

The communications subsystem of a network provides a medium for the exchange of information between host computers. The communications subsystem serves as the foundation on which more sophisticated conversations (often called "protocols"), between host computer entities (i.e., executing programs) may be implemented. Protocols have been developed to provide a wide variety of network services and functions, including:

- User access to other network hosts (and their resources)
- Movement of files from one host computer to another
- Network electronic mail

NSW is a protocol-oriented system, augmenting and extending the functions provided by existing network protocols.

Single host computers are often viewed abstractly as a collection of computing resources which may be parceled out to computing "tasks". In the course of a "task's" lifetime, many different computing system resources may be required to complete a given computational "task"; the following are representative examples:

- Storage and retrieval of files
- Program execution (of programs stored as files)
- Use of special peripherals (e.g., printers, tape drives, emulators, etc.)

Single host operating systems provide a uniform interface for accessing, and controls for mediating access to, computer system resources. If more than one host entity (e.g., program or user) can access and/or use a given resource (e.g., file, printer, etc.), that resource is said to be "shared". Many advantages accrue from resource sharing, including:

- Greater utilization through multiplexing (e.g., printers)
- Reuse without duplication (e.g., one copy of a compiler accessed by all users)
- Communication (e.g., between users)
- Lower cost

Similarly, computer network can make (selected) resources of all network host computers available to the network (user) community; i.e., resource sharing can occur across, as well as within the confines of host computers.

Now, the pioneering nature of NSW becomes apparent. NSW has succeeded in achieving a uniformity at the network level which had, in the past, only been accomplished within a single-host environment. To achieve uniformity and transparency of operation, NSW provides:

- A single standard command language
- A global, distributed file system
- Uniform access to and control of network resources (i.e., programs called "tools" as well as files)

- Integrated project management facilities
- Centralized accounting

All of the above has been achieved in a "heterogeneous" networking environment of diverse and incompatible host computers.

## **2.1 History**

Evolution of the NSW concept and development of the NSW system has occurred in several, sometimes overlapping, phases. The five (5) phases of NSW development [3] are briefly summarized below:

1. **Feasibility Demonstration (July 1974 through November 1975):** This phase included formulation of the basic NSW architecture, ad hoc implementation of the major software "components" (see Section 4.4.1) and demonstration of the following:
  - Use of IBM 360 batch tools
  - Use of TENEX interactive tools
  - Transparent file movement and translation, and
  - Rudimentary project management capabilities
2. **Detailed Component Design (June 1975 through March 1976):** During this period external and functional specifications were completed for the following major components: MSG (the NSW interprocess communication facility), Works Manager, Foreman and File Package. A minimal Front End (component) specification was also prepared during this period.
3. **Prototype Implementation (January 1976 through November 1977):** This phase was primarily devoted to implementation of NSW components for varying host environments. During this period:
  - TENEX, IBM and Multics MSG implementations were completed and demonstrated
  - Initial implementation of all TENEX components (Works Manager, Foreman and File Package) were completed and demonstrated
  - Implementation of the Multics Foreman and File Package components progressed to the point that a "rudimentary Multics interactive tool" [4] could be demonstrated
  - Development of the IBM Foreman and File Package components was started

- Preparation of the "Interim Reliability Plan" was initiated
4. Reliability and Performance Enhancement (January, 1977 through December 1980): This phase concentrated on:
- Completion of the "Interim Reliability Plan" and implementation of specified reliability scenarios
  - Addressing performance problems visible in the existing (primarily TENEX/TOPS2-20) component implementations through:
    - . Development of performance measurement routines (completed in February of 1978)
    - . Instrumentation of all (TENEX) components (completed in May of 1978)
  - Completion of the Foreman and File Package components for the IBM and Multics hosts.
5. Productization (January 1979 through April 1981): The NSW productization phase, covered by this report, is characterized by the introduction and implementation of a plan for managing the development and operation of the NSW system [5]. This plan formalized procedures for:
- Release integration and hand-off
  - Quality assurance
  - Software Trouble Report (STR) processing
  - Configuration management
  - Production operation
- . This phase can also be characterized as a period of unprecedented use and hardening of the NSW software. Many problems and deficiencies in the areas of operability, reliability, performance, user/operator interfaces, etc were successfully addressed. (Thus, the "Productization" phase is also considered to be a continuation of Phase 4: "Reliability and Performance Enhancement"). Also, in preparation for Phase 6 ("Technology Demonstration and Transfer Feasibility"), much of the planning for the AFLC Technology Demonstration was completed during this period. During the next, Technology Demonstration, phase selected AFLC sites will make use of and assess NSW and networking technology relative to current and future AFLC needs.

Phase 6 development activities will be based on current planning activities aimed at determining the NSW features and performance parameters required for the planned Technology Demonstration.

## 2.2 Policy Statements

The major near-term goals of the NSW project are summarized below using excerpts from policy statements issued during the contract period covered by this report:

### Project Evaluation Policy (November 24, 1979) [6]:

- "Our immediate goal should be to overcome known performance obstacles and rectify known design and feature deficiencies to provide a system capable of supporting a technology demonstration within the AFLC, serve as a repository and testbed for JOVIAL and ADA programming tools, and be used by the NSW development community and by selected members of the ARPA community."
- "The initial task should be a reevaluation of the functional performance characteristics desired in the system".
- "Our performance goal should be to provide all of the critical performance characteristics required by the above postulated user community, and, within our resource limitations, those desirable characteristics which will enhance user satisfaction."
- "Our implementation goal should be to have any redesign/re-implementation occur as a series of staged releases evolving from Release 4.1."

### Product Policy (November 24, 1979) [7]:

- "In general, ... proper attention shall be given to operability, [and] maintainability, in addition to functional performance."
- "The system ultimately provided shall permit tool execution which approximates the native environment performance, as well as in selected cases provide an appropriate execution environment."

### AFLC Demonstration Policy (November 24, 1979) [8]:

- "The intent is that the AFLC Demonstration have a high project priority"
- "The goal is to demonstrate selected aspects of NSW functionality. The effort will be executed in three phases, the first two of which provide introduction to networking and the NSW, and configure the necessary tool support for phase three. During the third phase, the functional demonstration of the NSW will occur using a selected AFLC task."
- "The demonstration will have functionality as its goal and help to define the ultimate product characteristics for the AFLC application."

NSW Support Strategy (December 26, 1979) [9]:

- "[A]. . . strategy for the use, support and modification of NSW release 4.1" requesting coordinated action on the part of PDC and ACC (see Section 3.1) to:
  - . Develop "a plan for the use and support of the current release 4.1"
  - . Participate "in regularly scheduled meetings of the 'working group' that has been set-up to coordinate the details of the AFLC Technology Demonstration."
  - . Ensure "a smooth transition of their [ACC and PDC] planning activities" to "assure . . . a smooth transition from NSW version 4.1"

### **3.0 NSW Project Organization**

During the prototyping and feasibility demonstration phases of NSW development (prior to January of 1979) the absence of project structure and discipline had been the norm. However, the desire to harden, ruggedized and enrich (i.e., "productize") the NSW system, combined with the increasing number of geographically distributed contractors, established the need for formal project management discipline. RADC responded to this need by preparing the "NSW Management Plan", which:

1. Establishes ". . . an overall approach for managing the development and operation of the National Software Works (NSW)" [10], and
2. Identifies "responsibilities of participating organizations, types of services to be provided, and directions for future growth" [11].

### 3.1 Organizational Structure

The roles of the six (6) primary NSW organizational entities, discussed in the "NSW Management Plan", are summarized below:

- Policy Group (PG): Largely staffed by individuals from the sponsor organizations (RADC and ARPA), PG responsibilities include resolution of "top level issues which impact NSW's present service stance and future range of applicability" [12]. Specific PG responsibilities include:
  - Strategic planning
  - Resource allocation, and
  - Organizational conflict resolution
- Architecture Control Contractor (ACC): ACC is responsible for:
  - "Ensuring [the] continued specification-level integrity of the NSW architecture" [13]
  - Interpreting the strategy for and orchestrating the development of NSW, and
  - Integrating the various software components delivered by the Development Maintenance Contractors (DMC's) into an NSW product which is both operable and usable.
- Product Development Contractor (PDC): "Focusing on those steps which can be taken, . . . to bring about near term, measurable improvements in either the feature, performance or reliability domains" [14], PDC is largely responsible for "making NSW a palpable, usable, [and] acceptable product" [15].
- NSW System Operations (NSWOPS): NSWOPS' role is to provide the NSW user community with "reliable, dependable and efficient computational services and comprehensive, correct and responsive information services" [16].
- Development/Maintenance Contractors (DMC's): The individual DMC's are responsible for implementation and/or maintenance of one or more NSW software components (see Section 4.4.1). DMC activities are

directed by specifications and/or guidelines established and maintained by ACC.

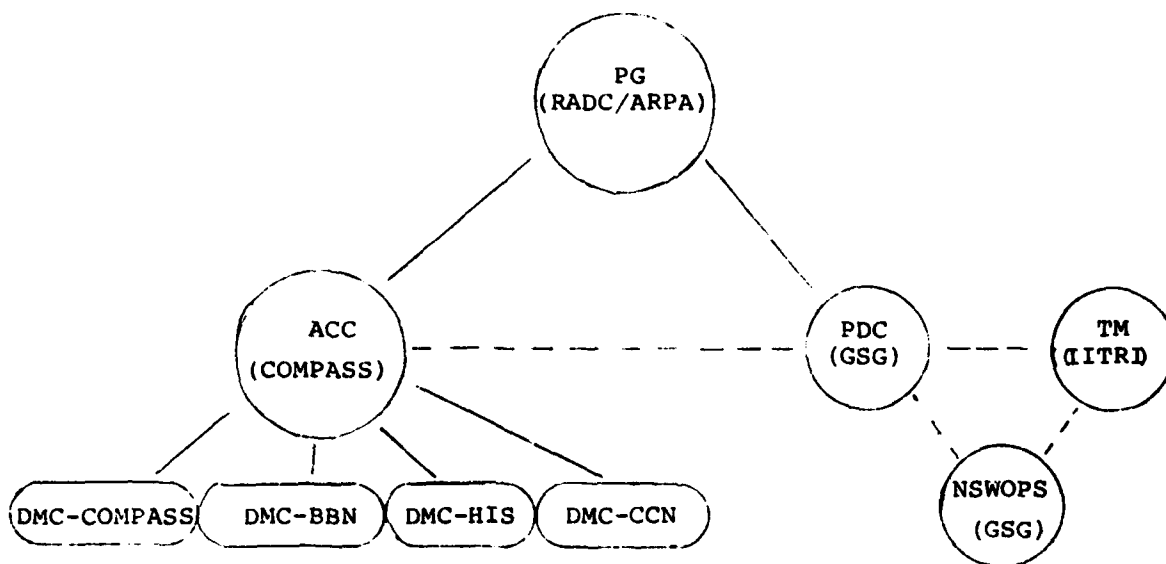
- Tool Manager (TM): The NSW Tool Manager is responsible for managing and supporting the selection, acquisition, installation and maintenance of (new) NSW tools (see Section 6.1.3 for a discussion of NSW tools).

### 3.2 NSW Contractors

Summarized below are the sponsors and/or contractors (and principal contact) responsible for each of the NSW organizational roles discussed in Section 3.1:

- Policy Group (PG)
  - RADC/ISCP: Dick Metzger, and Al Barnum
  - ARPA/IPTO: Larry Druffel
- Architecture Control Contractor (ACC)
  - Massachusetts Computer Associates (COMPASS): Charley Muntz
- Product Development Contractor (PDC)
  - General Systems Group, Inc.: Doug Payne
- NSW System Operations (NSWOPS)
  - General Systems Group, Inc.: Doug Payne
- NSW Tool manager (TM)
  - IIT Research Institute (IITRI): Loraine Duval
- NSW Development/Maintenance Contractors (DMCs)
  - Massachusetts Computer Associates (COMPASS): Charley Muntz
  - Bolt, Beranek and Newman, Inc. (BBN): Rick Schantz
  - UCLA/Campus Computing Network (CCN): Neil Ludlam
  - Honeywell Information Systems (HIS): John Ata
  - Specific component responsibilities of each DMC are detailed in Section 4.4.2.

Although no formal reporting structure has been established, Figure 3-1 (which is based on existing protocols and organizational interactions) provides a realistic approximation of NSW's organizational structure (dashed lines indicate significant organizational interactions which span the organizational hierarchy).



NSW Organizational Structure

Figure 3-1

#### **4.0 NSW System Architecture**

The (sub)sections which follow present a "bottom-up" overview of NSW system concepts and architecture. Separate sections are devoted to the following topics:

The ARPA Network

NSW host families

NSW software components

Participation of hosts in NSW systems configuration

#### 4.1 The ARPANET

NSW has been implemented in the ARPA Network (ARPANET) environment. A brief description of the ARPANET follows:

- "The ARPANET is an operational, computerized, packet switching DoD digital network which provides a capability for terminals or geographically separated computers, called hosts, to communicate with each other. The host computers often differ from one another in type, speed, word length, operating system, and other characteristics. Each terminal or host computer is connected into the network through a small local node computer called an IMP or TIP. The complete network is formed by interconnecting the IMPs through wideband communication lines (normally 50,000 bits per second) supplied by common carriers.

Each node is programmed to receive and forward messages to neighboring nodes in the network. During a typical operation, a host passes a message to its node; the message is passed from node to node through the network until it finally arrives at the destination IMP, which in turn passes it along to the destination host. This process normally takes less than 250 milliseconds.

Hosts communicate with each other via regular messages. A regular message may vary in length from 96 to 8159 bits, the first 96 of which are control bits called the leader. The leader is also used for sending control messages between the host and its IMP or TIP (node). The remainder of the message is the data or text.

For each regular message, the host specifies a destination, consisting of node, host, and handling type. These three parameters uniquely specify a connection between source and destination hosts. The handling type gives the connection specific characteristics, such as priority or non-priority transmission. Additional leader space has been reserved for a fourth parameter, to be used in future internetwork addressing. For each connection, messages are delivered to the destination in the same order that they were transmitted by the source.

For each regular message, the host also specifies a 12-bit identifier, the message-ID. The message-ID, together with the destination of the message, is used as the "name" of the message. The node uses this name to inform the host of the disposition of the message.

Therefore, if the host refrains from re-using a particular message-ID value (to a given destination) until the node has responded about that message-ID, messages will remain uniquely identified and the host can retransmit them in the event of a failure within the network.

After receiving a regular message from a host connected to it, a node breaks the message into several packets (currently the maximum data bits per packet is 1008) and passes these through the network in the direction of the destination. Eventually, when all packets arrive at the destination, they are reassembled to form the original message which is passed to the destination host. The destination node returns a positive acknowledgement for receipt of the message to the source host. This acknowledgement is called a Ready for Next Message (RFNM) and identifies the message being acknowledged by name. In some relatively rare cases, however, the message may not be delivered due to a node failure, line disruption, etc., in such cases an Incomplete Transmission message will be returned to the source host instead of a RFNM. In this case the message which was incompletely transmitted is also identified by name.

If a response from the destination node (either RFNM or Incomplete Transmission) is not delivered to the originating host, this condition will be detected by the source node, which will automatically inquire of the destination node whether the original message was correctly received and repeat the inquiry until a response is received from the destination node. This inquiry mechanism is timeout-driven, and each timeout period may vary between 30 and 45 seconds in length.

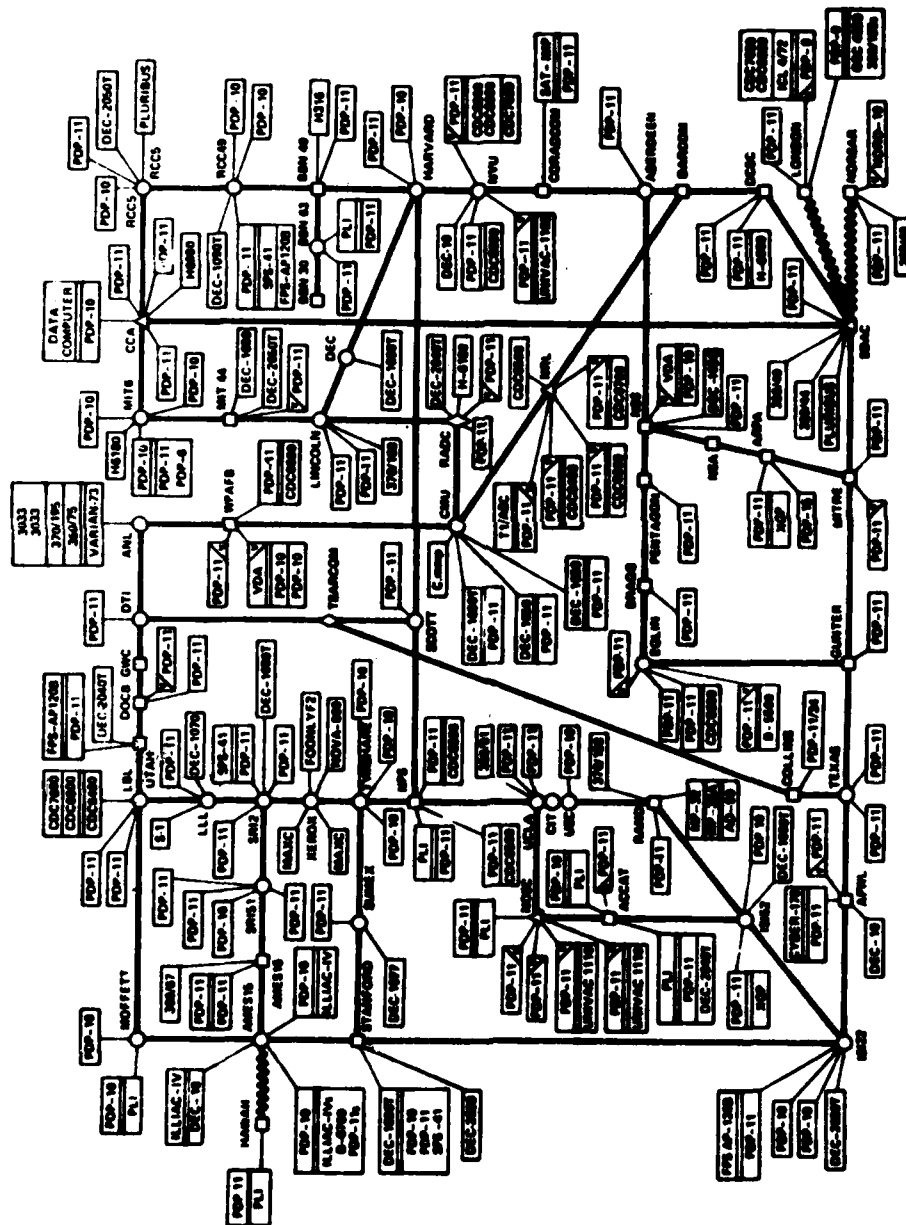
When a message arrives at its destination node, the leader is modified to indicate the source host, but the message-ID field is passed through unchanged. Thus, in addition to providing message identification between a host and its local node, the message-ID can provide a means for hosts to identify messages between themselves.

Users of the ARPANET may access local or distant SERVER computer (hosts) over the network. They may also exchange messages, create realtime links between users, transfer files from one computer to another, and submit batch jobs to distant computers. For a more complete description of these processes, see the ARPANET Protocol Handbook available from the [Network Information Center] NIC or the National Technical

Information System (NTIS), Springfield, VA. 22161 as  
AD A052594." [17]

The ARPANET communications subsystem configuration (i.e., IMPs  
and TIPs -- see above) is shown in Figure 4-1 [18]; Figure 4-2 is  
an ARPANET schematic which includes host identifiers [19].

# ARPANET LOGICAL MAP, MARCH 1980



PLEASE NOTE THAT WHILE THIS MAP SHOWS THE MOST  
POPULATION OF THE NETWORK ACCORDING TO THE  
BEST INFORMATION AVAILABLE, NO CLAIM CAN BE  
MADE FOR ITS ACCURACY.  
MOST COMPUTER COMMUNITIES OFFICE OF THE  
NETWORK INFORMATION CENTER  
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY)  
HOST NAMES

- IMP
- PLURIBUS IMP
- △ PLURIBUS TP
- ◇ PLURIBUS TP
- ▭ SATELLITE CIRCUIT
- ▲ VERY DISTANT HOST

Figure 4-1

# ARPANET GEOGRAPHIC MAP, MARCH 1980

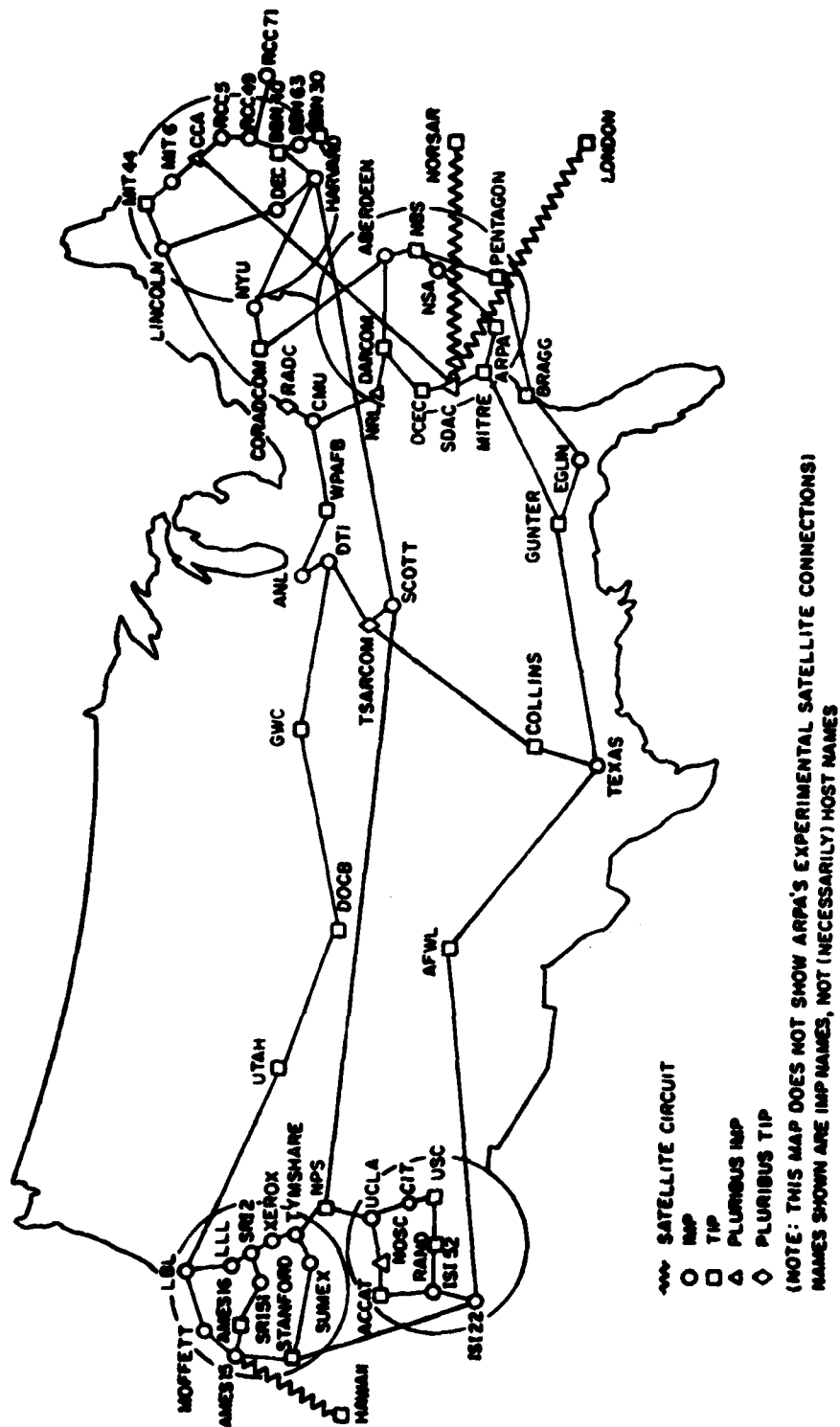


Figure 4-2

#### 4.2 NSW - A Network Operating System

The National Software Works (NSW) is an operating system. Because it has been implemented in a networking environment, NSW is often called a "network operating system". Much like a single host operating system, NSW manages network resources, providing its users with uniform access to files, programs (called "tools") and hardware resources available from network hosts which "participate" in the NSW system. However, the NSW network operating system differs from single-host operating systems in many significant and important ways:

- NSW operates in the networking environment, where interactions between two or more geographically dispersed computers may be required to complete a user's request.
- NSW is an operating system which, itself, has been built "on top" of existing single host operating systems.
- NSW has been built to operate on a set of DISSIMILAR computer systems, and to provide users with uniform, controlled access to the resources of these dissimilar computer systems.

### 4.3 Host Families

NSW software components (see Section 4.4.1) must be implemented for an ARPANET host computer system in order for that host to "participate" in a configuration of the NSW network operating system. Generally speaking, the level at which a host can participate in an NSW configuration is determined by the number of different NSW software components implemented for that host. Since the operability of NSW component implementations is largely independent of hardware configuration and/or operating system version/release, we often speak of "host families" when referring to NSW component implementations. That is, NSW software components may be implemented once, for a "host family", then the software components can be installed on each host (of that family) participating in the NSW system configuration.

The NSW host families and the current set of constituent ARPANET hosts (i.e., participating family members) are itemized below:

- TENEX/TOPS-20

ISIE: DECSys-20 (1090T) running TOPS-20 version 3A.

ISIC: DECSys-20 (KA10) running TOPS-20 version 3A.

RADC-20: DECSys-20 (2040T) running TOPS-20 version 4.

- IBM/OS

UCLA-CCN: 3033 running MVS (Note: CCN is currently converting from OS/VS1 to MVS)

- Multics

RADC-Multics: H6180 running Multics

- Unix

RADC-UNIX: PDP-11/45 running Unix (future Front End Host -- see Section 4.4.3)

ROBINS-UNIX: PDP-11/45 running Unix (future Front End Host -- see Section 4.4.3)

#### 4.4 NSW Software Architecture

The NSW functionality has been logically decomposed and grouped into a set of software "components". Host-independent specifications exist for each major NSW component. These specifications are used as the basis for realizing component implementations for different host families.

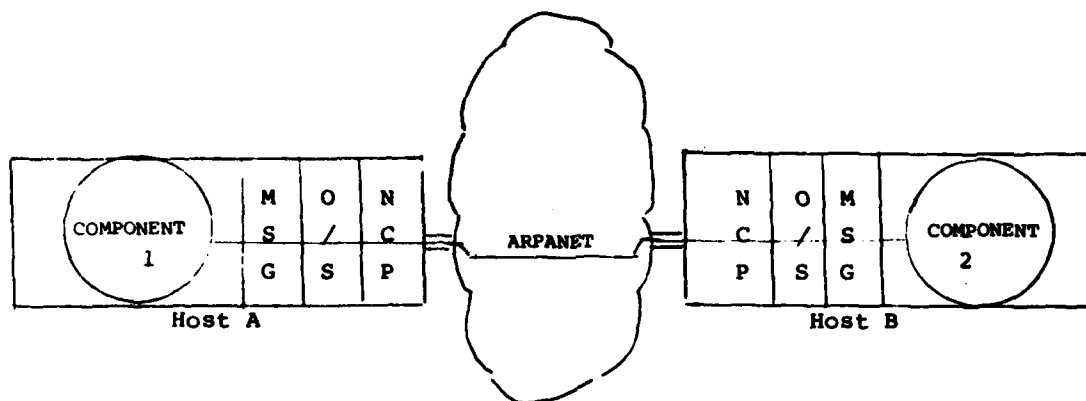
NSW components communicate with one another through an interprocess communication facility called MSG (Figure 4-3). MSG, combined with existing ARPANET "protocols", serves as the foundation for implementing higher level, NSW protocols. Protocol interactions between components are often grouped into "NSW scenarios"; each NSW scenario reliably implements one (or more) NSW operating system function(s).

For an ARPANET host to "participate" in an NSW system configuration, host (family) implementations of the interprocess communication facility (MSG) and at least one other NSW component must exist. The level at which a host (family) can participate in an NSW system is constrained by the set of NSW components implemented for that host (family). NSW software components (see Section 4.4.1) are each associated with a set of NSW operating system services. The level at which a particular network host participates in NSW system configuration is determined by the set of implemented services (executing components) the host provides for the NSW user community.

An NSW system "configuration" consists of a set of ARPANET hosts participating at various levels according to the NSW operating system services they provide (operating system services are implemented by the NSW software components - see above). An ARPANET host may participate in more than one NSW system configuration.

NSW resource management and control, including authentication, access control, synchronization and accounting are logically centralized in a set of "core system" software components. The "core system" components run on the "core system host"; there is one "core system host" per NSW system configuration. Thus, the "core system host" enjoys the highest level of participation in an NSW system configuration. Non-"core system" components, run primarily on non-"core system" network hosts but may (and frequently do) run on the "core system host" as well. A single "core system" host means that the cost for other non-"core system" hosts to participate in an NSW system configuration (in terms of system resources) is relatively low. That is, only the "core system host" pays the price to provide the expensive "core system" services mentioned above.

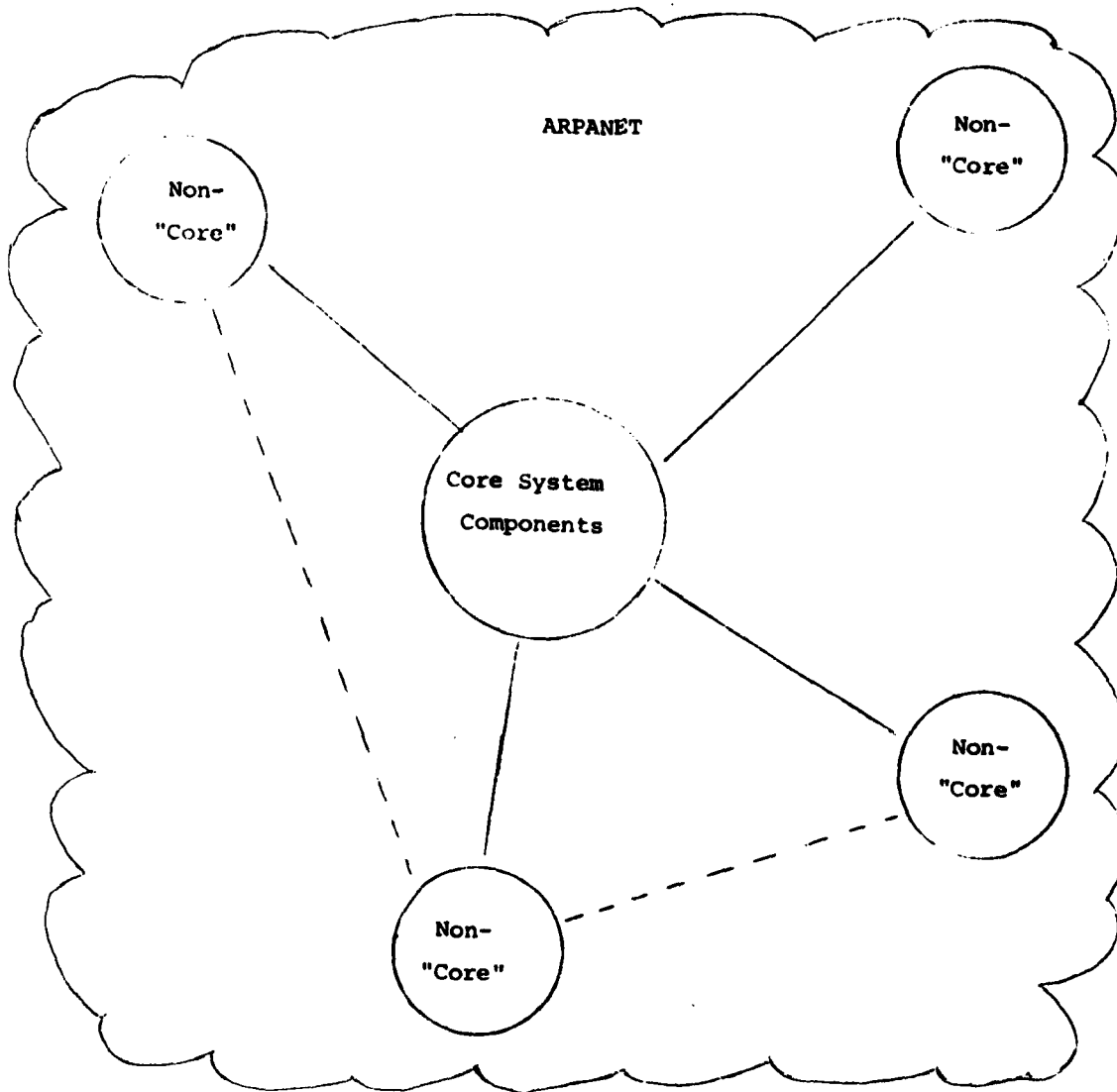
A central set of "core system" components also means that NSW logically takes the shape of a "star" (Figure 4-4). This star-like control structure does not preclude direct communication between non-"core system" components (as indicated by the dashed lines of Figure 4-4). However, such interactions are usually coordinated in advance with the "core system".



Protocol Layers	COMPONENT 1			COMPONENT 2		
	HOST A	MSG	(NSW interprocess communication)	HOST B	MSG	
	HOST A	O/S	(Host operating system)	HOST B	O/S	
	HOST A	NCP	(Network Control Program - network interface)	HOST B	NCP	
ARPANET COMMUNICATIONS SUBSYSTEM						

Component Communication Via MSG

Figure 4-3



NSW's Star Shaped Control Structure

Figure 4-4

#### 4.4.1 NSW Software Components

Logically speaking, the NSW system consists of eleven (11) functional software packages called "components". These components consist of generic (i.e., implementation independent) groups of NSW functionality which may be realized for one or more of the NSW host families (by NSW Development/Maintenance Contractors (DMC's) -- see Section 3.1). A brief description of each generic NSW component is given below:

- MSG: The interprocess communication facility used by all components to communicate with and request services of one another is called MSG.
- "Core System" Components:
  - Works Manager (WM): The Works Manager is NSW's system resource manager and control point, providing centralized authentication, access control, synchronization, and accounting. The Works Manager maintains several critical databases, including:
    - A central file catalog
    - A central tool (executable program) catalog
    - Project organizations (see Section 6.1.1)
    - User ids, passwords, and rights
  - Checkpointer (CHKPTR): The Checkpointer maintains Works Manager database integrity through periodic back-up.
  - Works Manager Operator (WMO): The WMO directs and coordinates the execution of "batch jobs" with non-"core system" Batch Job Package components (see below).
  - Fault Logger (FL): The Fault Logger is the central operator-visible collection point for "faults" (abnormal and/or error conditions) reported by other NSW components.
  - Operator Utility (OPRUTL): OPRUTL is one of many operator utilities; it provides NSW operators with a number of high-level "core system" component manipulation and clean-up operations. OPRUTL has been elevated to component status

because some operator functions communicate with the "core system" components through MSG.

- Non-"Core System" Components"

- Foreman (FM): The Foreman component oversees the execution of interactive tools (executable programs). One important function of the Foreman is to intercept operating system calls (e.g., file references) and direct them to NSW or the local operating system, as appropriate.
- File Package (FLPKG): The File Package component manages a host-resident portion of NSW filespace. File Packages resident on different NSW hosts interact with each other to transfer files between different (host-resident) portions of NSW file space. The File Package is also responsible for translating files as they are moved between hosts which belong to different host families.
- Batch Job Package (BJP): Under direction of the Works Manager Operator the Batch Job Package maps user-initiated "batch jobs" into the local host execution environment and oversees their execution.
- Front End (FE): The NSW Front End implements and interprets the NSW command language. Alternative Front End implementations may realize syntactically and semantically different command languages (see, for example, Section 6.2.1). A request initiated by the Front End always begins an NSW "scenario", involving interactions with at least one other NSW component. Based on user-supplied commands and parameters, the Front End requests services implemented by other NSW components (usually the "core system" Works Manager).
- Dispatcher (DSPCHR): Contacting a standard host socket causes the Dispatcher to place the (prospective) NSW user in contact with a spawned Front End instance.

#### 4.4.2 DMC Responsibilities

The four (4) NSW component Development/Maintenance Contractors (see Section 3.2) are each responsible for several host family component implementations. These responsibilities are summarized below by host family.

##### TOPS-20

MSG - BBN

##### "Core System" Components:

- Works Manager - COMPASS
- Checkpointer - COMPASS
- Works Manager Operator - COMPASS
- Fault Logger - BBN
- Operator Utility - COMPASS

##### Non-"Core System" Components:

- Foreman - BBN
- File Package - COMPASS
- Front End - COMPASS
- Dispatcher - BBN

##### IBM

MSG - UCLA/CCN

##### "Core System" Components:

\*\* none \*\*

##### Non-"Core System" Components:

- Foreman - UCLA/CCN
- File Package - UCLA/CCN

- Batch Job Package - UCLA/CCN

#### Multics

MSG - HIS

"Core System" Components:

\*\* none \*\*

Non-"Core System" Components:

- Foreman - HIS
- File Package - HIS

#### Unix

MSG - BBN

"Core System" Components:

\*\* none \*\*

Non-"Core System" Components:

- Front End - BBN

#### 4.4.3 Participation of Hosts in NSW Configurations

The level at which an ARPANET host may participate in an NSW system configuration is determined by the collection of software components implemented for the associated host family. (Note: An implementation of MSG is required for any level of participation.) The five (5) disjoint levels at which a host may participate in an NSW configuration (or, alternatively the five sets of operating services a host may offer by participating in an NSW system configuration) are each described below:

- Core System Host (CSH): Only one host per NSW configuration can serve as the "core-system host" by providing the following NSW functions: authentication, access control, synchronization and accounting. Host family implementations of the following "core system" components are required:
  - works Manager
  - Checkpointter
- . Works Manager Operator (required only if the NSW configuration includes one or more Batch Job Package components).
- . Fault Logger (required only if the NSW configuration includes one or more components which report "faults" to the Fault Logger).
- . Operator Utility (as required by host family implementations of the Works Manager and Checkpointer).
- File Bearing Host (FBH): Any host for which a (host family) implementation of the File Package component exists can oversee a portion of NSW filespace resident on that host by providing support for:
  - . File transfer and translation between host-resident portions of NSW filespace
  - . The movement of files between NSW filespace and the local host file system.
- . The Datacomputer (now defunct) is an example of an ARPANET host which might wish to participate in an NSW system configuration only as a File Bearing Host (i.e., not as a Tool Bearing Host -- see below).

- (Interactive) Tool Bearing Host (TBH): Any host for which a host family implementation of the Foreman component exists can provide support for the execution of interactive "tools" (executable programs). However, the Foreman cannot, without a File Package component, support tools which reference NSW files. Thus, participation as a File Bearing Host is almost always considered a prerequisite to participation as a Tool Bearing Host.
- Batch Job Host (BJH): Any host for which a host family implementation of the Batch Job Package component exists can participate in an NSW system configuration as the executor of NSW batch job processing on that host. Of course, such participation assumes a "core system" configuration which includes a Works Manager Operator component (see above). Also, because certain information required for batch processing is transmitted from the Works Manager Operator to the Batch Job Package through the NSW file system, support for NSW files (i.e., a host-resident File Package component) is also required.
- Front End Host (FEH): A Front End Host acts as a user access point for an NSW system configuration. Any host for which a host family implementation of the Front End component exists can be an NSW access point providing user interface (i.e., NSW command language) support.

#### 4.4.4 Levels of Host Family Participation

Using the "levels of participation" discussed in Section 4.4.3, we can summarize the levels of participation possible for each NSW host family identified in Section 4.3:

- TOPS-20 (TOPS-20 host can participate in an NSW configuration as one or more of the following):

- Core System Host (CSH)
- File Bearing Host (FBH)
- Tool Bearing Host (TBH)
- Front End Host (FEH)

##### IBM:

- File Bearing Host (FBH)
- Tool Bearing Host (TBH)
- Batch Job Host (BJH)

##### Multics:

- File Bearing Host (FBH)
- Tool Bearing Host (TBH)

##### Unix:

- Front End Host (FEH)

One or more "levels of participation" have been assigned to each host family listed above. An ARPANET host which belongs to (i.e., is compatible with) one of the NSW host families CAN participate at any of the levels listed above for that family. For example, any ARPANET DECSys-20 running TOPS-20 can participate in an NSW system configuration as a:

- Core System Host (CSH)
- File Bearing Host (FBH)
- Tool Bearing Host (TBH)

- Front End Host (FEH)

or any combination of the above. The RADC-20 ARPANET host, a DECSys-20 running TOPS-20, provides a specific example. Currently the RADC-20 participates in several NSW configurations. In the NSW User System configuration (see Sections 4.4.5 and 5.1), the RADC-20 participates as a Tool Bearing Host and a File Bearing Host, but not as a Core System Host or a Front End Host.

#### 4.4.5 NSW User System Configuration

The production NSW configuration available to NSW users is called the "NSW User System". The User System configuration (Figure 4-5) consists of ARPANET hosts participating at various levels (see Section 4.4.3). The level(s) of host participation in the User System configuration are summarized below (by ARPANET host):

##### USC/ISIE (TOPS-20)

- Core System Host
- Tool (and File) Bearing Host
- Front End Host

##### USC/ISIC (TOPS-20)

- Tool (and File) Bearing Host
- Front End Host

##### RADC-20 (TOPS-20)

- Tool (and File) Bearing Host

##### UCLA-CCN (soon to be running MVS)

- Tool (and File) Bearing Host
- Batch Job Host

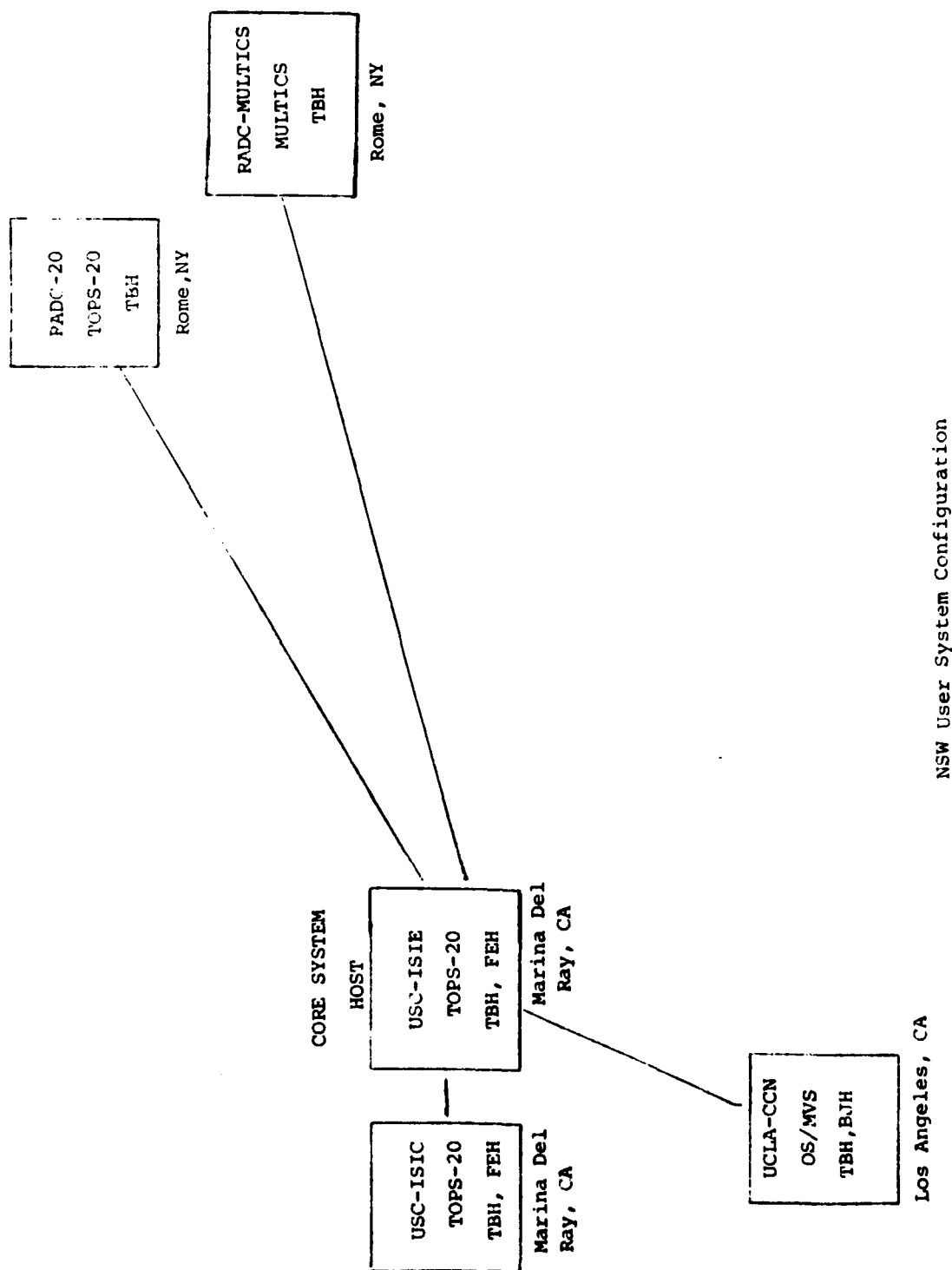
##### RADC-MULTICS

- Tool (and File) Bearing Host

Note the following:

- \* USC-ISIE is the only core system host in the User System configuration.
- \* RADC-20 could be, but is not, a Front End Host (even though a Front End component implementation exists for the TOPS-20 host family).
- \* UCLA-CCN is the only Batch Job Host (the IBM host family is the only family for which a Batch Job Package has been implemented).

- \* UCLA/CCN and RADC-MULTICS could not be Front End Hosts (Front End components have not been implemented for either the IBM or Multics host families).
- \* The User system configuration includes no Unix hosts (the Unix Front End has not yet been released for integration into the User System configuration).



NSW User System Configuration

Figure 4-5

## 5.0 NSW System Development

At the highest level, NSW system development can be viewed as a sequence of stages, each of which is performed and controlled by a different NSW organization. The four (4) stages of the NSW system development life cycle are briefly described below (see Figure 5-1):

1. **Component Development and Maintenance:** The NSW Development/Maintenance Contractors (DMC's -- see Section 3.1) are responsible for developing and maintaining host family implementations of specific NSW software components. When a new component version is considered ready for release, the responsible DMC delivers it to the NSW Architecture Control Contractor (ACC) for integration.
2. **System Integration:** It is ACC's job to integrate components received from the various DMC's into an operable NSW system and to verify that the components interact properly with one another. The DMC's work with ACC to isolate and correct problems; components which do not "pass muster" are returned to the appropriate DMC for additional work. Since the DMC's operate asynchronously from one another, NSW system integration is, defacto, an incremental process. That is, at most a few components are introduced into a stable NSW system configuration at a time. However, incremental integration has the advantage of making ACC's integration testing (and problem isolation) task somewhat easier. When ACC is confident that the integrated NSW system is ready for production operation, its constituent components are packaged into a system release which is then delivered to the NSW Product Development Contractor (PDC) for Quality Assurance.
3. **Quality Assurance (Q/A):** It is PDC's task to verify, through independent testing, that NSW system releases are ready for
  - Production operation by NSW System Operations (NSWOPS), and
  - Use by the NSW user community.
- Upon completion of Q/A, PDC provides the NSW Policy Group (PG) with recommendations regarding production operation of the release, including any corrective action which may be required to make the release

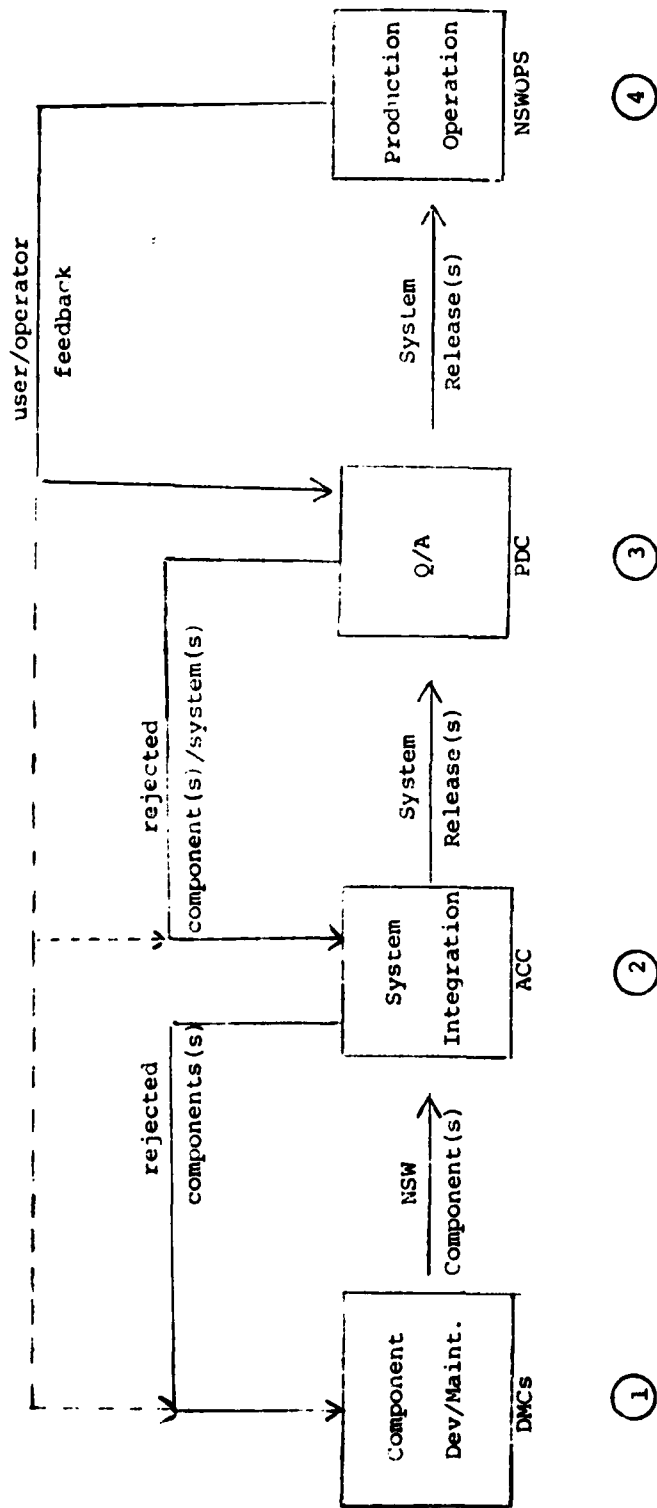
operable/usable. Based on recommendations from PDC as well as input from ACC, and NSWOPS, PG makes one of two determinations:

- Return the release to ACC (and the DMC's) for further work, or
- Deliver the release to NSWOPS for installation and production operation.

4. Installation and Production Operation: NSWOPS operates the national NSW User system configuration on behalf of the NSW user community. When a new release (or notification thereof) is received from PDC, NSWOPS:

- Schedules an installation date
- Broadcasts a summary of the release (including release date) to the NSW user community, and
- (On the designated date) installs and begins operation of the new release.

Note (see Figure 5-1) that each stage of the NSW system development process provides a feedback loop to the immediately preceding stage. In this way, user and operator experiences in the form of bugs, deficiencies and/or desired features may be communicated to PDC, ACC and ultimately the DMC's for analysis, prioritization and action.



Stages of NSW System Development

Figure 5-1

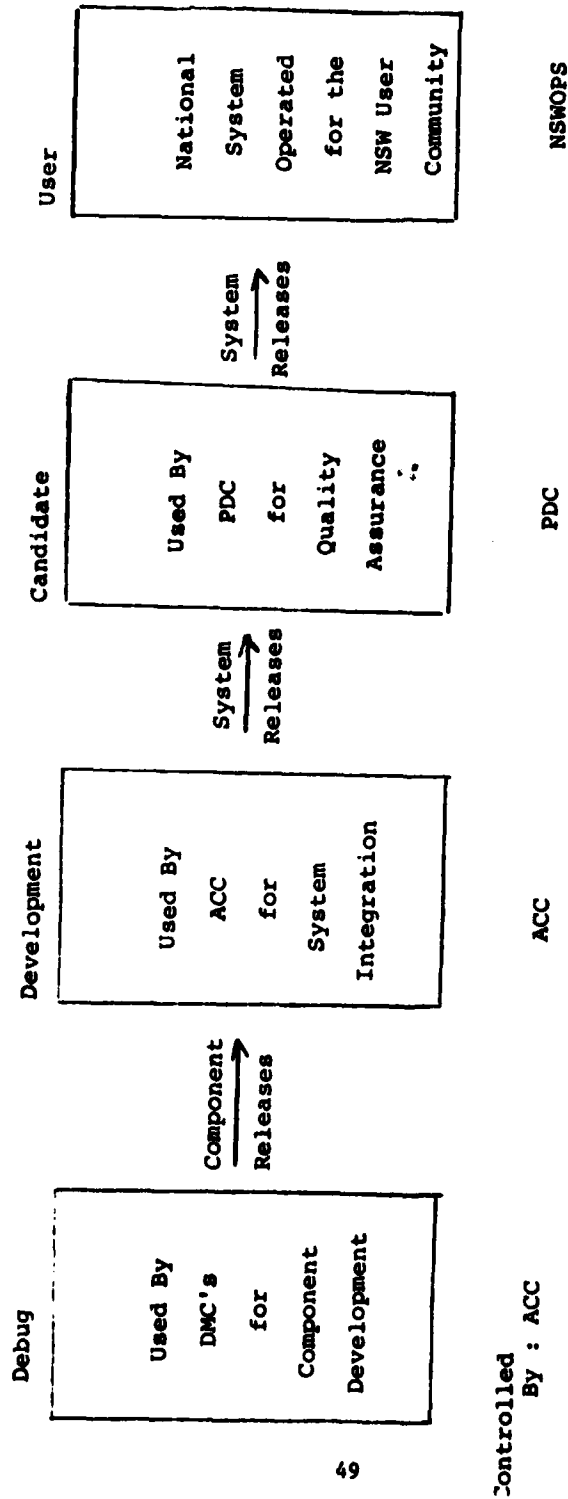
### 5.1 NSW System Configurations

To maximize efficiency, it must be possible for each step of the NSW system development process to proceed independently of all others. In particular, it must be possible for:

- NSWOPS to operate the current release while new releases are at various stages of development.
- PDC to perform Q/A on a new release (n+1) while ACC is integrating a future release (n+2) and NSWOPS is operating the currently installed release (n).
- ACC to integrate a release (n+1) while PDC is performing Q/A on a recent release (n) and DMC's are preparing components for future integration (release n+1 or greater).
- DMC's to prepare new component version(s) independent of ACC integration activities.

Four (4) distinct NSW system configurations were established to provide the necessary organizational autonomy which would allow each step of NSW system development to proceed independently of all others (Figure 5-2):

1. The NSW "Debug" System configuration, controlled by ACC, is used by the DMC's to prepare new component versions.
2. The NSW "Development" System configuration is controlled, operated, and used by ACC to integrate components into system releases which can then be delivered to PDC for Q/A.
3. The NSW "Candidate" System configuration, operated by NSWOPS, is controlled and used by PDC for Q/A.
4. The NSW "User" System configuration is controlled and operated by NSWOPS on behalf of the NSW user community.



NSW System Configuration

Figure 5-2

## 5.2 NSW Testing Hierarchy

NSW System software is separately tested during each stage of the system development process (see Section 5.0). Tests for a given stage are most often applied to one or more NSW components which have been installed in the associated NSW system configuration (see Section 5.1). Devise and applied tests are oriented toward the system development activities of each stage (e.g., Q/A) and reflect progress along the development continuum leading to production operation of new NSW system releases. Collectively, tests applied during development stages one (1) through four (4) form a hierarchy in which individual components are tested first (this is called "unit" testing), then individual components are integrated into a full NSW system configuration, which receives two separate batteries of tests: integration and Q/A. The testing activities associated with each stage of the NSW system development process are summarized below:

1. Unit Testing: Each NSW software component implementation is referred to as a unit. These tests are (often) applied to components installed in the NSW Debug System configuration. Unit tests are devised and applied by the DMC with implementation responsibility to:
  - Components in isolation to verify correct operation of internal functions, and/or
  - Small collections of components to verify that the component interacts properly with other NSW components.
- . Components which have "passed" unit testing are delivered to ACC for integration.
2. Integration testing: It is ACC's responsibility to:
  - Integrate NSW components received from DMC's into an operational NSW system,
  - Verify the system, specification and inter-component integrity of complete component configurations (i.e., full NSW system configurations), and
  - Package and deliver NSW releases to PDC for Q/A
- . Integration tests are devised and applied to components (received from the DMC's) which have been installed in the NSW Development System. Due to the asynchronous

nature of component deliveries from the DMC's, NSW system integration is largely an incremental process. However, incremental integration does have its advantages: if the installation of new components in the Development System can be spaced (in time) sufficiently apart from one another, the job of isolating problems (when they do occur) becomes significantly easier. ACC has, for this reason, adopted the incremental integration and testing approach. Incremental integration does require continuous operation and support of the NSW Development System which has the additional advantage of providing NSW developers (DMC's) with access to a fully-integrated, operational NSW system prior to Q/A and release for production operation. When the integration process has been completed, ACC packages the current set of components, databases and documentation for delivery to PDC who will perform Q/A.

3. **Quality Assurance (Q/A) Testing:** PDC is responsible for independent validation and verification of the new NSW systems released to ACC. During the Q/A testing, NSW systems delivered by ACC are installed and operated by NSWOPS in the NSW Candidate System configuration. PDC prepares and applies independent system-level tests to each major new NSW system release. The results of these tests are communicated to the development community (ACC) and, along with recommendations for action, to PG for determination of release disposition. (A more detailed discussion of GSG's quality assurance may be found in Section 5.4.)
4. **Production Operation:** NSWOPS installs and operates new NSW system releases for the NSW user community in the NSW User System configuration; these releases are delivered to NSWOPS by PDC (as directed by PG). In addition, NSWOPS is responsible for the timely resolution of user questions and problems (sometimes called Software Trouble Reports (STRs) -- see Section 5.5). Thus, NSWOPS must view and review the NSW system from both the operational and user perspectives. To provide an NSW system which is maximally operable and trouble-free, the testing efforts of stages one (1) through three (3) of the system development process concentrate on removing as many (major) problems and deficiencies as possible. Loosely speaking, the testing an NSW system receives during production operation is "use", which is undoubtedly the most important kind of testing and an important source of feedback for PDC, ACC and the DMC's as well. User (and operator) acceptability is

the ultimate test. (For a detailed discussion of GSG's system operation responsibilities and activities see Section 7.0.)

### 5.3 NSW System Release and Assessment Procedure

When system integration (and testing) has been completed, ACC packages and delivers the new NSW system to PDC for Quality Assurance (Q/A). Due to the autonomous operational requirements of the IBM and Multics NSW hosts, this DELIVERY takes the following form organizationally:

ACC: Delivers all TOPS-20 components to PDC/NSWOPS

NSWOPS: Installs and operates NSW TOPS-20 component configurations (including the "core system" components -- see Section 4.4) on the USC-ISIE, USC-ISIC and RADC-20 hosts

DMC-CCN: Installs and operates an NSW IBM component configuration on the UCLA-CCN host

DMC-HIS: Installs and operates an NSW Multics component configuration on the RADC-Multics host.

When a new system has been delivered for Q/A by ACC, a release assessment and transition procedure developed by PDC (and ACC) takes effect. This procedure covers the last two stages of the NSW system development process:

Stage 3: Quality Assurance by PDC

Stage 4: Production Operation by NSWOPS

As contractor for both the PDC and NSWOPS organizations, GSG is responsible for these two stages of the NSW system development process. Note that the procedure outlined below is presented from the PDC/NSWOPS perspective (i.e., that of PDC receiving releases for Q/A from ACC and that of NSWOPS receiving releases from PDC for production operation). The steps of the NSW system release and assessment procedure are:

1. ACC: Integrate NSW components received from the DMC's into an operational NSW system; perform integration testing.
2. ACC: When system integration has been completed:
  - Identify and obtain specific versions for each "generic configuration item" (see Section 5.6 and Appendix C) of the new NSW system, and

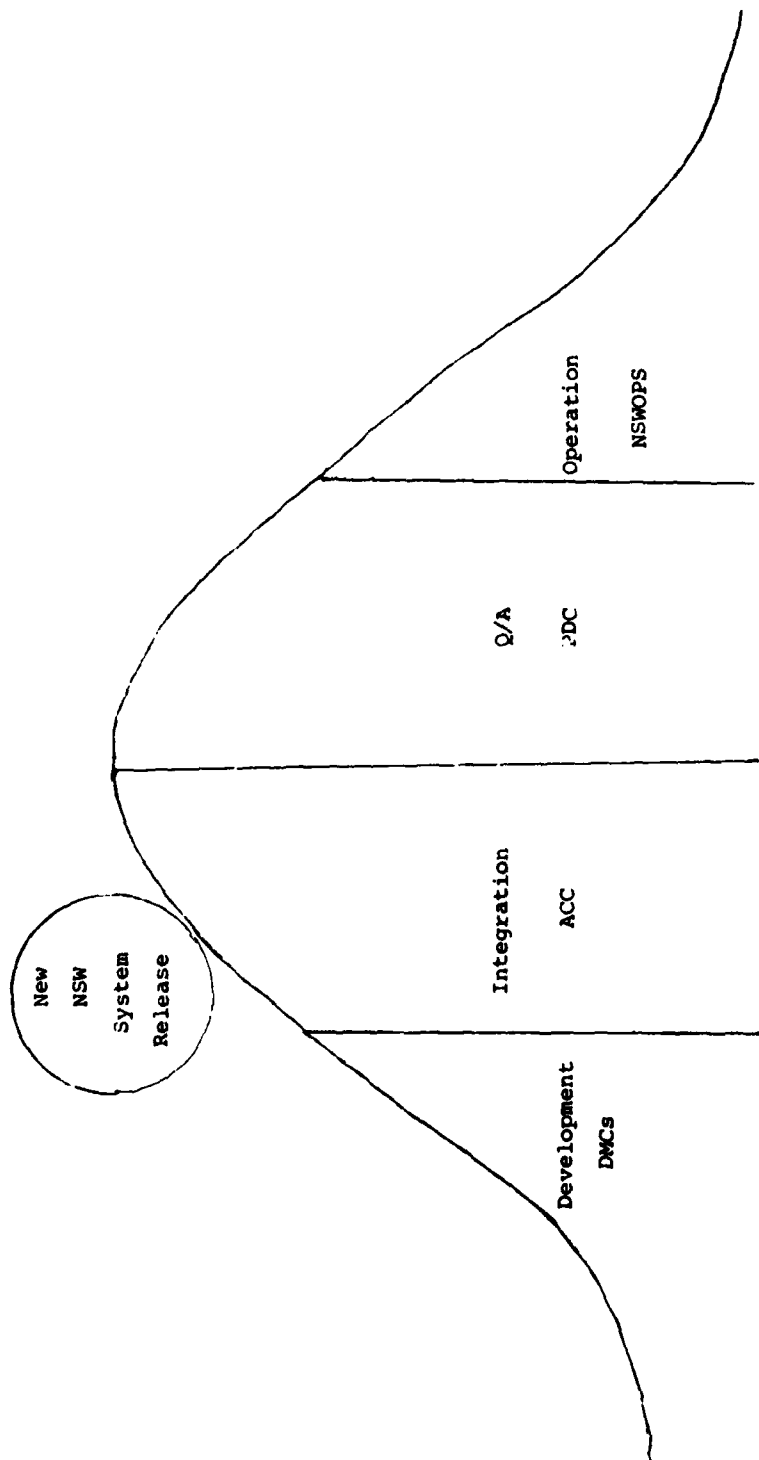
- Deliver the NSW System and TOPS-20 Host Family "packets" (of configuration items - see Section 5.6) to PDC/NSWOPS.
3. NSWOPS: Install selected configuration items from the "NSW System" and TOPS-20 "Host Family" packets in the NSW Candidate System configuration; coordinate installation of the IBM Host Family and Multics Host Family packets with DMC-CCN and DMC-HIS respectively.
  4. ACC: Verify that the new NSW system has been installed properly in the NSW Candidate System configuration; coordinate verification activities with DMC-CCN and DMC-HIS
  5. NSWOPS: Operate the new NSW system in the NSW Candidate System configuration (including the "core system") for PDC during Quality Assurance; direct and coordinate operational activities with DMC-CCN and DMC-HIS.
  6. PDC: Direct the application of quality assurance tests to the new NSW system release (see Section 5.4 for details); report results and release recommendations to PG and ACC.
  7. PG: Based on Q/A test results and input from PDC, NSWOPS and ACC, select one of the following courses of action:
    - a. Return the release to ACC for major modification (i.e., return step 1), or
    - b. Fix major problems and release to NSWOPS; i.e.:
      1. ACC: Direct DMCs to correct selected bugs and deliver new components for Candidate System installation ASAP.
      2. PDC: When new components have been received from ACC and/or selected DMCs, verify (if possible) that all major problems have been corrected and no significant, new problems have been introduced.
      3. Proceed with Step 8
    - c. Release to NSWOPS (i.e., proceed with Step 8)

8. NSWOPS: As directed by PG, prepare for, install and operate the new NSW system release on behalf of the NSW user community; direct and coordinate installation/operation activities with DMC-CCN and DMC-HIS.

Delivery of a new NSW system release to PDC for O/A is a significant event in the life cycle of an NSW system (see Figure 5-3). It signals:

- Completion of a major amount of the work associated with a given release, and
- Beginning of work "in earnest" on the next major NSW system release.

A further reason is that, as a result of PDC O/A activities, it is most often the case that PG elects to have ONLY major problems corrected (option b of step 7 above) before releasing the system to NSWOPS for production operation.



NSW System Development Life Cycle

Figure 5-3

#### 5.4 Quality Assurance Testing

PDC (GSG) has responsibility for assuring the quality of releases submitted by ACC. This role is as close as any on the NSW project to "independent validation and verification", a familiar DoD term. PDC's Q/A tests represent the most extensive and complete testing a new NSW system receives before it is released to NSWOPS for production operation. PDC's Q/A testing activities have evolved into the following logical sequence of steps:

1. Prepare and distribute a "Test Plan" for the new NSW system release (see for example, the "NSW 5.0 Test Plan" [20]) which identifies:
  - a. The current set of "generic configuration items" (see Section 5.6)
  - b. The status of all outstanding STR's (see Section 5.5) including specific tests prepared for each STR addressed by the new system release
  - c. The NSW host and software component configurations to be used for the release
  - d. The tests to be applied (see below), and
  - e. A schedule for completing testing activities.
2. Update existing Q/A test procedures, scripts, etc., and devise new tests as necessary (concurrent with Step 1).
3. Apply Q/A tests (PDC and NSWOPS) to the new release (i.e., the "Test Plan" is carried out under PDC's direction.)
4. Prepare and distribute a "Test Summary" of the Q/A testing activities (see for example, the "NSW 5.0 Test Summary" [21]), which includes:
  - a. An overview of Q/A test results and recommendations
  - b. A summary of significant events, which occurred during the test period, which impacted Q/A testing
  - c. A summary of the results for each type of testing (e.g., regression, STR-specific, etc. -- see below).

- d. An annotated list of new STR's uncovered during Q/A testing.

The "Test Summary" is distributed by PDC to PG and ACC. On the basis of the "Test Summary" and input from PDC, ACC and NSwOPS, PG determines the future disposition of the new NSw system, as discussed in Section 5.3.

The Q/A tests applied to each new NSw system are of several different types, each designed to test different integration levels of NSw functionality. These tests are not exhaustive, but they do represent a significant assessment of the NSw functionality provided the user. With the exception of STR-specific tests, Q/A testing methodologies have been prepared for each type of testing, as described below:

1. STR-Specific Testing: Prior to receipt of a new NSw system from ACC, PDC prepares tests for each Software Trouble Report (STR -- see Section 5.5) which will be addressed by the new release. These tests are used to verify that problems have been corrected and that no new problems or side-effects have been introduced in the process. Also, STR-specific tests are used to assess new and/or improved NSw functionality. These tests are documented in the "Test Plan" which is distributed prior to the beginning of the Q/A testing period (i.e., prior to receipt of a new release from ACC).
2. Regression Testing: A methodology has been devised for regression testing new NSw releases. If a system function or feature, which worked properly in the preceding release, no longer functions correctly in the new release, a "regression" is said to have occurred. Regression tests are designed to uncover these differences. However, the NSw regression tests are more general in that they have been designed to uncover as many differences as possible between successive NSw releases, including:
  - Improvements
  - Regressions, and
  - Problems which persist from one release to another
- Regression tests consist of a number of distinct test scripts. These scripts are composed of NSw command sequences which are applied to the user interface (exercising user-visible NSw functionality). Both application of regression test scripts (to the user

interface) and collection of NSW system responses have been automated. The steps of the NSW regression testing procedure are summarized below:

1. Update regression tests as necessary to reflect differences between the successive releases.
  2. Use automated means to apply test scripts to the new NSW system and collect responses.
  3. Compare (automatically) the responses for each test script to those retained from the preceeding release, and summarize the incremental differences.
  4. Analyze the differences (obtained in step 3) to identify improvements, regressions and problems exhibited by both the new and immediately preceeding releases.
  5. Prepare and submit new STR's to ACC for each regression identified.
- . This procedure is simple, yet powerful (identifying major differences between successive releases) and efficient (highly automated).
3. File Transformation Testing: File transformations are the vehicle which NSW provides for data residing on one NSW host to be converted into a form which may be used by a "tool" (see Section 6.1.3) residing on an NSW host of a different "family" (see Section 4.3). File transformations allow the output of one tool to be used by another tool of a different family. Thus, NSW file transformations are the key to success of tool interactions between NSW host families (e.g., the use of a tool followed by the use of a tool which resides on a host which belongs to a different (incompatible) family). Conversion of an EBCDIC character file residing on an IBM host to an ASCII character file during movement to a TOPS-20 host is a simple example of an NSW file transformation. File transformations embody record structure, format effectors, as well as word/character encodings. Because the NSW concept is highly dependent on these transformations, a "Methodology and Plan for Testing File Transformation" [22] was developed. Due to the nature of the differences between NSW host families, the distributed nature of the NSW file system (and its protocols), and the data translation approach

adopted, NSW file transformations are perhaps one of the most difficult packages of NSW system functionality to test. The file transformation testing methodology is applied to new NSW system releases to verify the fidelity of this functionality. A simplified version of the basic procedure is outlined below:

- a. Update the "Methodology and Plan for Testing NSW File Transformations" as necessary to reflect incremental changes from the preceeding release.
  - b. Prepare and/or modify file transformation test case files, as necessary; import these files into the NSW Candidate System.
  - c. Use automated means to force file transformations by moving test case files from one NSW host to another (these hosts may or may not belong to different host families).
  - d. Analyze resultant files to determine whether the file transformations under test have been successfully completed.
  - e. Prepare and submit STR's to ACC for each identified file transformation anomaly or problem.
4. Interactive Tool Testing: Tool testing is concerned with verifying the proper operation of "tools" (executable programs) installed in the NSW environment. Interactive tools are the most prevalent type of NSW tool (see Sections 6.1.3 and 6.2.3). Although tool testing is viewed largely as a Tool Manager responsibility (see Section 3.1), it is an area which GSG addressed during the NSW system development process. Tool operability is one of the most important concerns of NSW user community, because tool availability is the user's reason for using the NSW system. Therefore, GSG developed the "Interactive Tool Testing Methodology" [23] and applied it to a number of NSW tools delivered with NSW release 5.0. The results of these tests are summarized in the "Summary of Interactive Tool Tests: NSW Release 5.0" [24]. These tests will be applied as necessary to (selected sets of) the tools delivered with all future NSW releases.

## 5.5 STR Processing

When NSW users, contractors, etc uncover suspected problems with the NSW software, a Software Trouble Report (STR) is prepared to document the problem. In addition, STR's are used to report system deficiencies. An STR includes some or all of the following information:

- Originator (e.g., name of user or contractor)
- Description of the problem, deficiency or improvement
- Date and time of submission
- Type of problem (component, tool or documentation)
- Release number
- Urgency
- Etc.

STR's are a critical part of the NSW system development process, particularly configuration management (see Section 5.6) of successive NSW releases. Because of the pervasive use of STR's for reporting bugs, deficiencies and new feature requests, NSW system releases have largely become STR-directed. That is, the NSW system modifications integrated and packaged into a system consist largely of components delivered by DMC's which address a given set of STR's (including deficiencies and improvements).

GSG, as Product Development Contractor (PDC), developed the procedures and protocols for the original STR accounting system. At that time, STR's were known as "NSW Standard Transactions" (or NST's) and the accounting system was known as the "NST System". The NST System was designed to be general and highly flexible. NST's were envisioned as a convenient means for documenting and/or tracking many communications ("transactions") other than NSW system software bugs, including:

- Deficiencies
- Improvements
- Inquiries (questions)

- Operations procedures
- Regularly generated reports
- Etc.

Eventually, the large number of NST's and the informal, unregulated protocol for interaction between NSW organizations became a major burden to users of the NSW System. Consequently, a tool for monitoring STR's, called MONSTR was developed by Massachusetts Computer Associates for use by all NSW contractors, and (eventually) NSW users. MONSTR serves as a STR repository and implements a more rigid, table-driven protocol for the movement of STR's between NSW organizations on the road to resolution. Besides implementing the protocol for resolution of STR's, MONSTR provides a number of other facilities including:

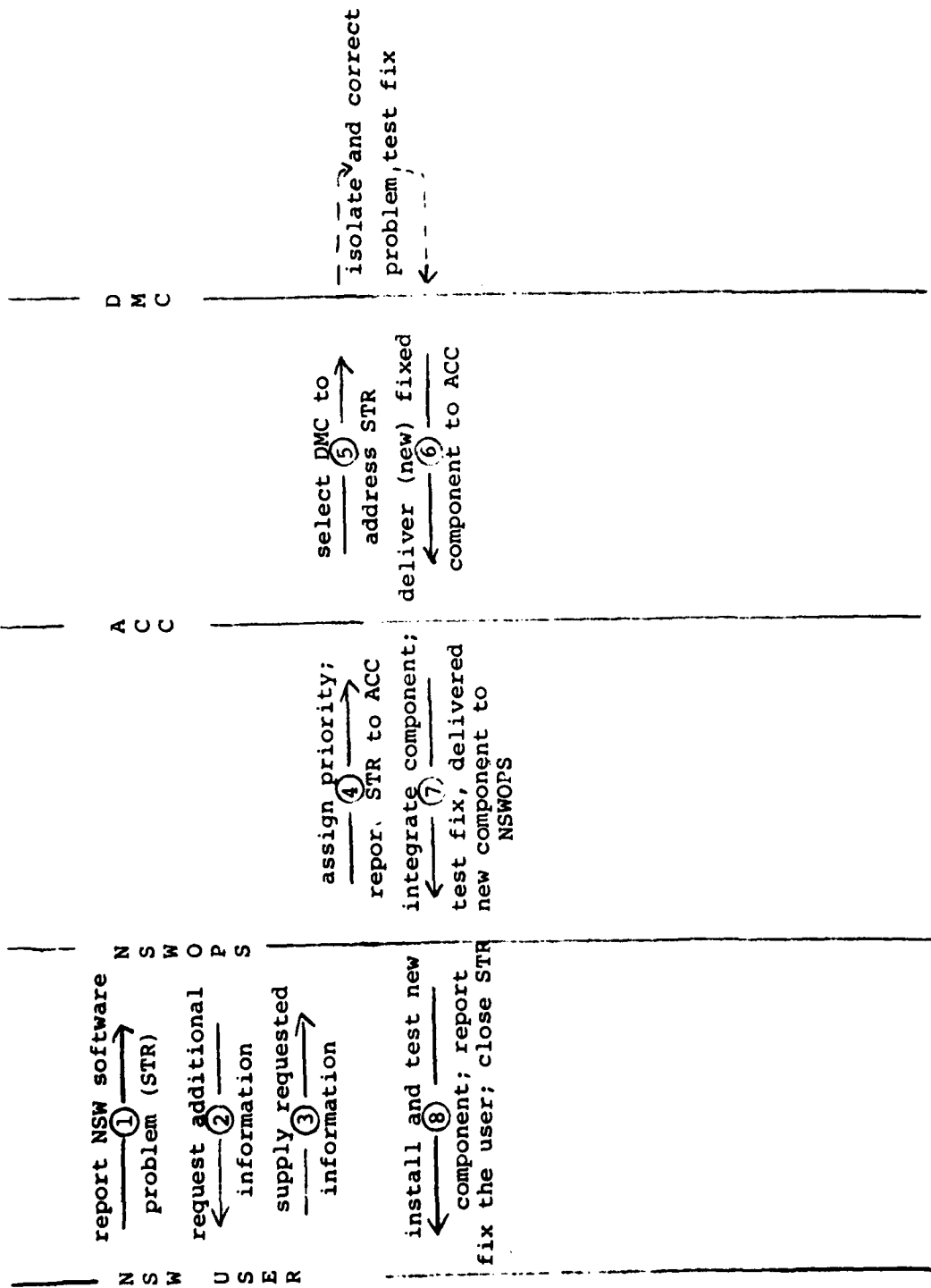
- Archiving of terminated STR's
- Generation of STR reports by STR attribute (e.g., by STR identifier, priority, responsible organization, etc.)
- Etc.

To understand the role STR's play in the NSW system development process, consider the following example (Figure 5-4):

1. An NSW user creates an STR, which reports a suspected NSW software problem to NSWOPS.
2. After reviewing the user's STR, NSWOPS decides that the problem is legitimate (i.e., neither user misunderstanding nor error), but decides to request additional information (e.g., a typescript) from the user.
3. The NSW user responds by supplying NSWOPS with the information (typescript) requested.
4. NSWOPS assigns a priority for resolution of the STR and dispatches the STR to ACC. (Note: For this example we shall assume that the problem is interfering dramatically with use of the NSW system and that a high priority has been assigned to its resolution).
5. ACC reviews the STR and dispatches it to the appropriate DMC for corrective action (Note: If more information were required or the STR were not actually a software problem, it would be returned to NSWOPS).

6. The DMC selected by ACC addresses the problem by preparing a new component version which is delivered to ACC for integration and testing; the STR is returned to ACC with an analysis of the problem and a description of the fix.
7. ACC integrates and tests the new component in a full NSW system configuration. If the new component properly addresses the STR, it is passed on to NSWOPS along with the STR (which includes a user-oriented description of the problem and how it was rectified). Otherwise, the component is returned to the appropriate DMC for additional work.
8. NSWOPS installs the component (PDC may elect to perform Q/A), notifies the user that the problem has been corrected, and "terminates" the STR.

The above example describes the means by which a high priority problem, substantially impacting use of the NSW system, would be resolved. Note, however, that STR's are normally addressed and integrated into major NSW system releases, which are, from the user's point-of-view, spaced many months apart.



Organizational Actions for STR Resolution: An Example

Figure 5-4

## 5.6 Configuration Management

Software configuration management (CM) is a discipline for installing software changes which allows the evolution of a software system to be tracked and controlled. By tracking and controlling software changes, CM becomes an integral part of a formalized system software life cycle planning and development process. Such formal development discipline becomes extremely important for projects with characteristics similar or identical to those of NSW, namely:

- A large number of geographically distributed contractors, who are
- working autonomously on a large number of constituent software components, which must, ultimately, be
- Integrated into a workable software product.

Software CM is an integral part of the plan for developing, maintaining and operating the NSW system ("The NSW Management Plan" [2]) The NSW project has collectively been working toward a CM implementation goal since the beginning of the contract period covered by this report. GSG's PDC organization received a majority of the charter and responsibility for developing NSW configuration management procedures.

Software CM is generally viewed as a continually evolving, three-step process:

1. Configuration identification
2. Configuration control
3. Configuration auditing

The CM approach adopted for the NSW project results from a collective effort on the part of both PDC and ACC. With suggestions and input from ACC (Massachusetts Computer Associates), GSG has prepared a "NSW Software Management and Control Plan" [26] which deals with each of the above CM activities in detail. A brief characterization of each CM activity as it pertains to the NSW system follows:

- \* **CONFIGURATION IDENTIFICATION:** This is the process of identifying the objects, called "items" (e.g., programs, data files, etc.) which are to be controlled. Each item, without a specific version, is called a "generic configuration item". When a release is packaged, a specific version is identified for each generic

configuration item to be included with the release; items which include version identification are simply called "configuration items" (CI's). PDC and ACC agreed (with concurrence from PG) that only major, aggregate items of the NSW system would be placed under configuration control. These aggregate items include:

- NSW software component executables
- NSW "tool" executables
- NSW system database skeletons
- Host-independent functional specifications
- User documentation
- Host-dependent component documentation
- Operator documentation

Note: This list of aggregate items does not, for example, include the constituent source and object modules for each component executable (see below).

- Configuration management of source and object modules for each software component (see Section 4.4.1) has been delegated to the DMC responsible for the component implementation. After a consensus had been reached on configuration item granularity, PDC developed a methodology for identifying, through hierarchical decomposition, the "generic configuration items" for the NSW system. The decomposition consists of collections of generic CI's called "packets". The approach taken was to first identify generic CI's which applied to more than one host family (i.e., the entire NSW system), then to identify generic CI's which apply to more than one software component, tool, etc of a host family (i.e., to the entire host family), then to identify the CI's which apply to individual host family component implementations and tools, respectively. The CI packets produced by this decomposition are:

1. NSW System Packet (CI's which apply to multiple host families)
2. TOPS-20 Host Family Packet, containing the following subpackets

- a. Generic TOPS-20 packet (CI's which apply to more than one TOPS-20 component or tool)
  - b. TOPS-20 component packet(s) (one for each TOPS-20 component implementation)
  - c. TOPS-20 tool packet(s) (one for each TOPS-20 tool)
- 3. IBM Host Family Packet
  - a. Generic IBM packet
  - b. IBM component packet(s)
  - c. IBM tool packet(s)
- 4. Multics Host Family Packet
  - a. Generic Multics packet
  - b. Multics component packet(s)
  - c. Multics tool packet(s)
- 5. Unix Host Family Packet
  - a. Generic Unix packet
  - b. Unix component packet(s)

(Note: There are currently no Unix-based NSW tools)

- Because of autonomous operational requirements, configuration management of the IBM and Multics Host Family packets (with the exception of configuration auditing -- see below) have been delegated to DMC-CCN and DMC-HIS, respectively. Generic configuration items for both the "NSW System" and "TOPS-20 Host Family" packets are enumerated in Appendix B of this report.
- NSW organizational and contractor responsibility for configuration management of the above packets (as installed in the various NSW configurations used for NSW system development) is summarized below:
  - Debug System: ACC (COMPASS)
  - Development System: ACC (COMPASS)

- Candidate System: PDC (GSG)

- User System: NSWOPS (GSG)

- \* **CONFIGURATION CONTROL:** Once configuration items have been identified and CM procedures implemented, a large part of the "fixed cost" associated with configuration management is done. The following disciplines are the major constituents used to control (incremental) NSW software changes:

1. **Autonomous Component Development:** DMC's are responsible for development and maintenance of individual NSW software components. ACC is largely responsible for directing DMC's to make software changes and upgrades. DMC's modify and deliver software components which reflect the changes requested by ACC or suggested by the DMC's. ACC completely controls integration of new component versions into fully operable NSW system configurations.
2. **Autonomous NSW System Configuration:** Associating a separate NSW system configuration with each major stage of the NSW system development process, and delegating control for each configuration to a single NSW organization (see Section 5.1) provides the mechanism for controlling changes to the NSW system during the NSW system development process.
3. **Release Packaging:** When preparing a system release for delivery to PDC, ACC is required to make all software changes known to PDC and NSWOPS. Software changes directed by ACC usually take one of two forms:
  - A fully specified enhancement which must be implemented by at least one DMC, or
  - An STR which documents a problem, deficiency or improvement (STR's are accessible to NSW contractors through MONSTR -- see Section 5.5).

Because ACC controls the integration process, it is likely that most NSW software changes are communicated to ACC by the DMC's, which can in turn be communicated to PDC prior to delivery of a new release for Q/A.

However, because of autonomous and asynchronous component development practices, it is possible for DMC's to make changes to components which

are never known by ACC. Since PDC/NSWOPS must rely on ACC for a complete and accurate picture of the incremental software changes which comprise any given release, we should note that this approach to configuration management is not foolproof. However, experience indicates the level of formality and control which has been achieved is sufficient to assure smooth evolution of the NSW system.

- \* **CONFIGURATION AUDITING:** Configuration auditing is the process by which information is obtained on the current state or past history of an installed set of NSW configuration items. One of the most important aspects of configuration auditing is the ability to identify differences between any two sets of NSW configuration items (e.g., system or component releases). Such procedures are extremely important in the NSW context because of IBM and Multics host family autonomy (see Section 5.1). For example, auditing allows ACC, PDC and NSWOPS to determine what changes (if any) have been made in the sets of IBM or Multics host (family) configuration items (installed in the NSW Development, Candidate or User system configurations) without maintaining control over the set of installed CI's. Configuration auditing procedures have been implemented and are regularly exercised in each NSW system configuration (i.e., the Development, Candidate and User Systems).

### 5.7 NSW Release History

Version 3.1 of the NSW system was installed in the NSW User System configuration just prior to the contract period covered by this report. Two (2) major and one (1) incremental release(s) occurred during the past contract period. The salient features of each release are outlined below:

NSW Version 4.0: A major NSW system release addressing several STR's and also including a few significant, but largely operator-visible (i.e., user-invisible) enhancements.

- Released to PDC for Q/A: May 2, 1979
- Major Improvements:
  - Support for TOPS-20 version 101B
  - First release of the Fault Logger component
  - New Checkpointer and TOPS-20 Foreman components
  - Improved operator interface
  - More robust Multics TBH
  - Separate descriptors for each tool executable
  - . Package logs (component compiling/linking prescriptions)
- STR's Addressed: Unknown
- Testing Completed by PDC: August 14, 1979
- STR fixes failing Q/A: Unknown
- New STR's identified: 300+ (action suggested by PDC on 128)
- Major (New) Problems:
  - Database integrity
  - . Improved component resilience (especially TOPS-20 MSG)

Protocol violations

Resource consumption

User interface anomalies

- Released to NSW Users: Never; upgraded to NSW 4.1.

NSW Version 4.1: An incremental upgrade of NSW release 4.0, addressing many of the problems mentioned above under NSW version 4.0.

- Released to PDC for O/A: September 18, 1979

- Major improvements:

Database integrity problems resolved

- . Component resilience (new TOPS-20 MSG and Foreman, and IBM Foreman component(s))

Several protocol violations rectified

- . Resource consumption (unnecessary component instances and activations) improved
- . User interface (error messages, control characters and connection handling) significantly upgraded

- STR's Addressed: 69

- Testing Completed by PDC: September 28, 1979

- STR Fixes Failing O/A: 9

- New STR's Identified: 49 (most of these were scheduled to be addressed by NSW version 5.0 -- see below)

- Major (new) problems:

Reliability scenario failures

TOPS-20 MSG link allocation problems

Local TOPS-20 MSG failures

Forcibly terminated components

Front End failures

Node size limitations

- Released to NSW users: October 5, 1979

NSW Version 5.0: A major, new, largely STR-directed NSW system release also concentrating on release transition and configuration management procedures.

- PDC "Test Plan": July 30, 1980
- Released to PDC for Q/A: July 31, 1980
- Major improvements:

Support for TOPS-20 version 4; more resilient TOPS-20 MSG

New TOPS-20 Foreman component (workspace creation, archiving and reliability scenarios)

User interface improvements (restored tools, Foreman workspace and file delivery prompts)

Multics TBH more robust

- STR's addressed: 65
- Q/A testing completed by PDC: August 29, 1980
- PDC "Test Summary": September 11, 1980
- NSWOPS "Interactive Tool Test Summary": November 14, 1980
- STR fixes failing Q/A: 8
- New STR's identified: 67
- Major (new) problems:

TOPS-20 MSG and Front End failures (most of the problems in this area were diagnosed and corrected before release of NSW 5.0 to users)

- Released to NSW users: October 20, 1980

## 6.0 NSW User Interface

In the first of two major sections which follow, we provide the reader with a user-oriented overview of NSW functions and capabilities. This first section sets the stage for the second section which summarizes and discusses:

- The current status of the NSW User System
- Progress (during the contract period covered by this report) in evolving the NSW concept, and
- Future directions dictated by known requirements, deficiencies and problem areas

### 6.1 Overview of NSW Capabilities

The NSW system operates in a networking environment and is analogous, in its functions and capabilities, to a single-host operating system. In particular, NSW implements a standard command interpreter, providing NSW users with uniform access to:

- Files resident in NSW's distributed file system
- Tools available on NSW hosts in the NSW system configuration.

In addition, NSW incorporates a built-in project management facility which provides project managers with a vehicle for delegating responsibilities and implementing informal protocols among project members. An overview of NSW's project management capabilities, file system, tools and command language are each covered by the individual subsections which follow.

### 6.1.1 Project Management

The project management facilities of NSW are an integral part of its design. Each new NSW user is assigned a "project", a "node" and a "password", which must be supplied during the NSW authentication sequence (i.e., LOGIN). Thus, a user wishing to use NSW facilities first logs in to the assigned node of his/her project. The user's login node establishes a context for the duration of the his/her NSW session.

NSW projects are collections of nodes. Much like the organizational structure of many companies, the nodes of a project form a hierarchy (also called a "project tree"). Each node must have a "supervisory" parent and may have zero (0) or more sons. We say that a given node was "created" by its parent and that a node (with the appropriate "right" - see below) may act as a "supervisory" parent by creating sons of its own. (Note: since a node establishes the context for a user's NSW session, the terms "node" and "user" may, at times, be used interchangeably). Nodes establish a context by acting as place-holders for tool and file access "rights". Associated with each node is:

- A list of tools which may be accessed (used) by that node, and
- A list of file "keys" controlling access to named (groups of) NSW files.

A node may receive rights from any "supervisory" parent in the tree of which the node is a direct descendent. A node may give any of its rights to any of one or more of its sons (or their sons, etc.).

NSW provides built-in project management tools for:

- Manipulating (creating and deleting) nodes (or sub-trees) of the project tree
- Assigning rights to and removing rights from a son node
- Examining nodes of the project tree and their tool/file access rights

These project management tools are governed by tool rights as well. For example, a user (node) wishing to create a (new) son node must have received rights for the "create node" tool from a "supervisory" parent.

To illustrate NSW project management capabilities, consider the following example. Initially, there is one node (node "A") in the project tree (see Figure 6-1) with:

Tool rights for:

- All NSW project management tools:

Create-node

Delete-node

Examine-node

Assign-rights

Remove-rights

- A small set of program development and documentation tools:

Teco (a text editor)

PL/I (load and go compiler)

Runoff (a text/document processor)

File rights:

A and B

- . (Note: A and B are called "keys" to the NSW namespace for files. NSW filenames consist of a number of "components" separated by periods ("."). For example, "A.PROJ.TEAM" is an NSW filename with separate components. The "keys" A and B may be used to access any NSW file whose name begins, respectively, with the component A or the component B. The notion of NSW file rights has been simplified somewhat for this example; for additional information, please refer to the NSW Users' Reference Manual (27)).

Using the create-node tool, node A creates nodes B, C and D, then, using the assign-rights tool, gives each new node the rights shown in Figure 6-2. The rights assigned by node A allows only node D to create new nodes (node D can not, however, delete son nodes). Also, nodes B and C have a more restricted set of file rights (i.e., "view" of NSW filespace) than node D. In fact, node D can access any file which nodes B or C can create or access, but because node D has the more general file right "A", it can also create or access files which nodes B and C can't

(e.g., files beginning with A.D). Continuing with our example, node D creates three new son nodes (DA, DB and DC) assigning each the rights shown in Figure 6-3. After surveying the project tree with the "examine node" tool (which can not be shown diagrammatically here), node A implements the following project management decisions:

- Node B will not be needed; node A uses the "delete-node" tool to delete node B (Figure 6-4)
- Node A notices that node D has been overzealous in creating new son nodes; node A uses the "remove rights" tool to revoke the "create-node" tool right from node D (Figure 6-5)
- Node DB needs temporary access to some NSW files which begin with B.A. Node D cannot access files beginning with B, so s/he requests assistance from node A. Node A temporarily assigns file right B.A to node DB (see Figure 6-6), but not to node D who doesn't need it.

In this example, we have examined the NSW project management capabilities for:

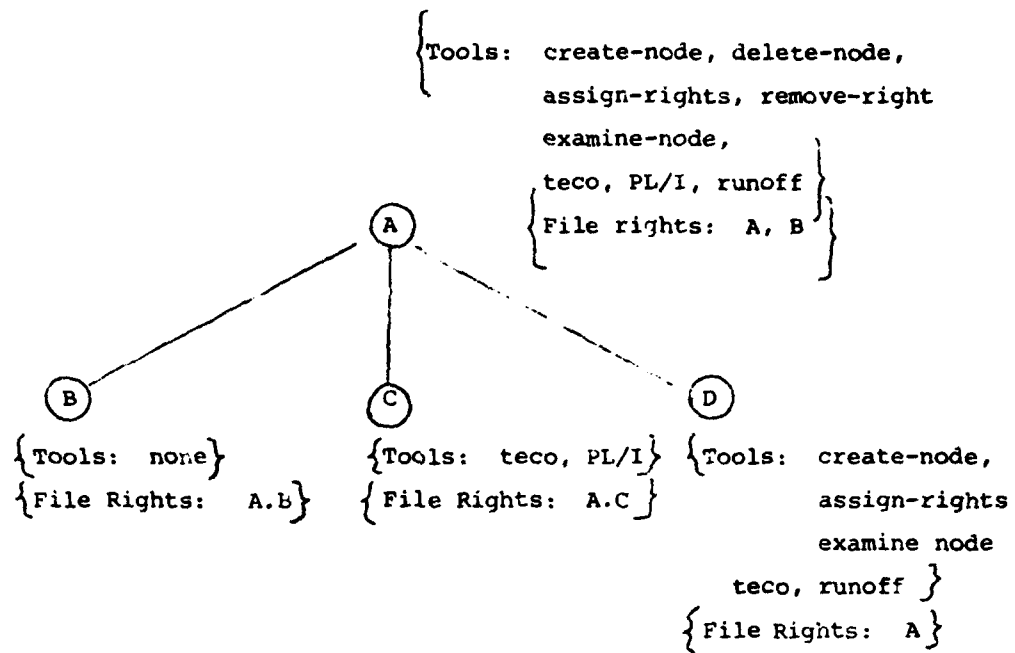
- Creating and deleting son nodes, and
- Assigning and removing tool and file rights.

We have also tried to provide feeling for the managerial actions which may be implemented by project directors and team members who use the NSW project management tools.

① { Tools: create-node, delete-node  
assign-rights, remove-rights,  
examine-node,  
teco, PL/I, runoff }  
{ File Rights: A, B }

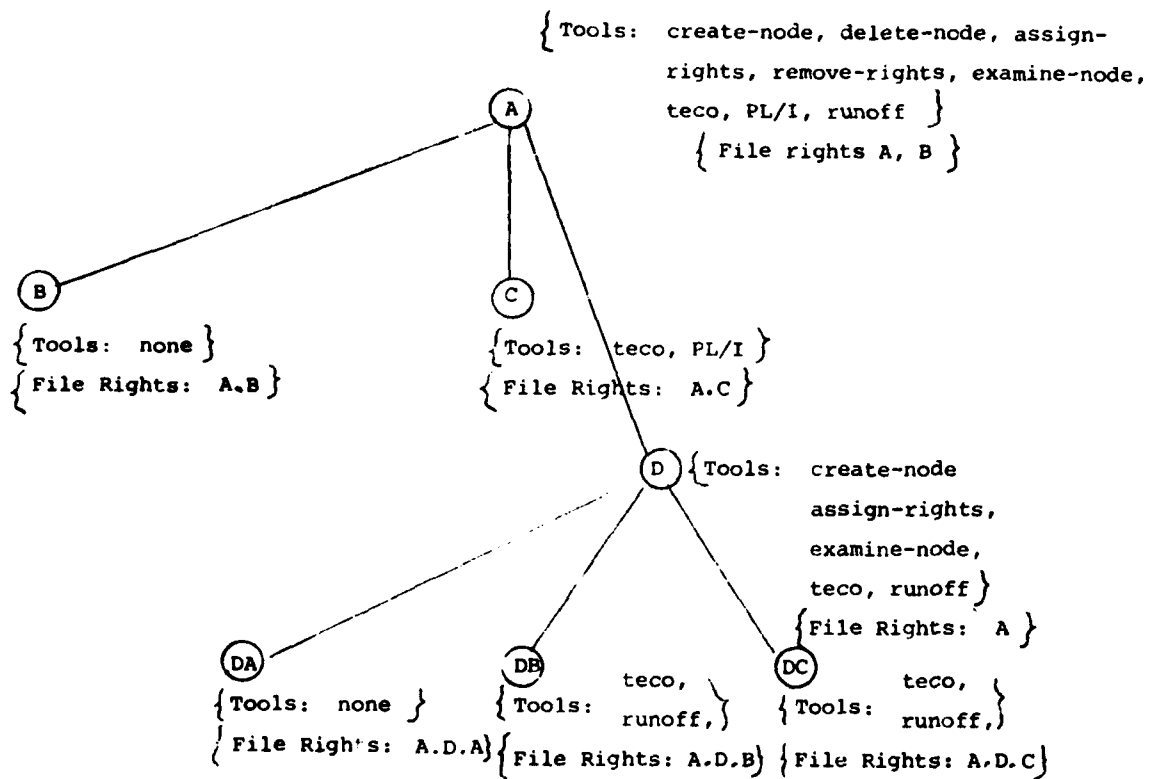
Project Node A and Tool/File Rights

Figure 6-1



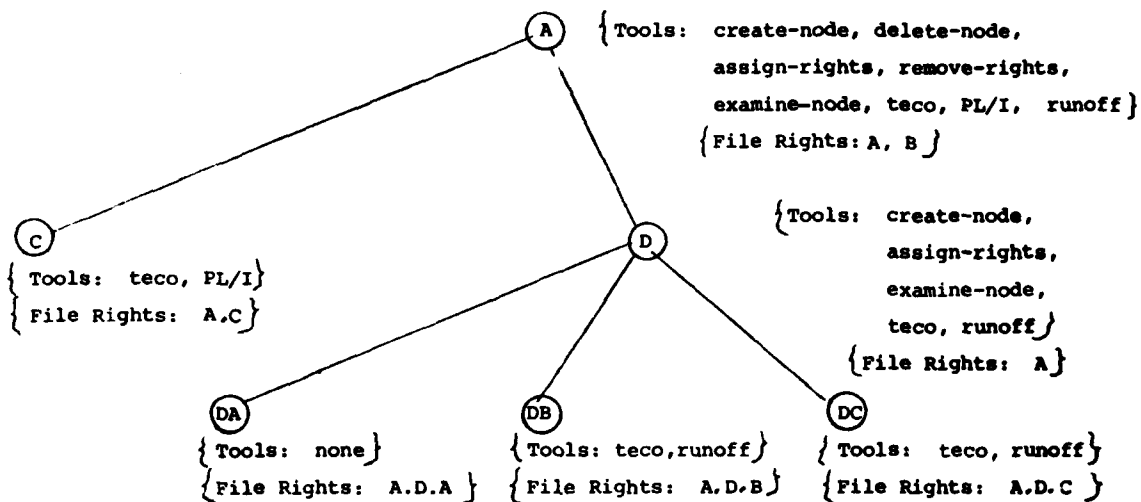
Node A Creates Nodes B, C, D and Assigns Tool/File Rights

Figure 6-2



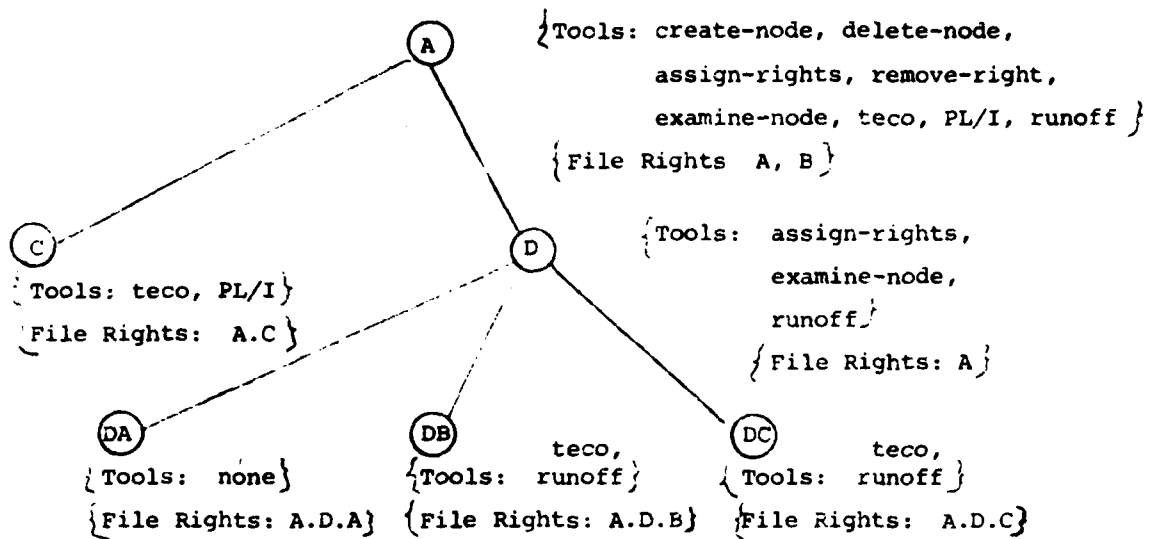
Node D Creates Nodes DA, DB, DC and Assigns Tool/File Rights.

Figure 6-3



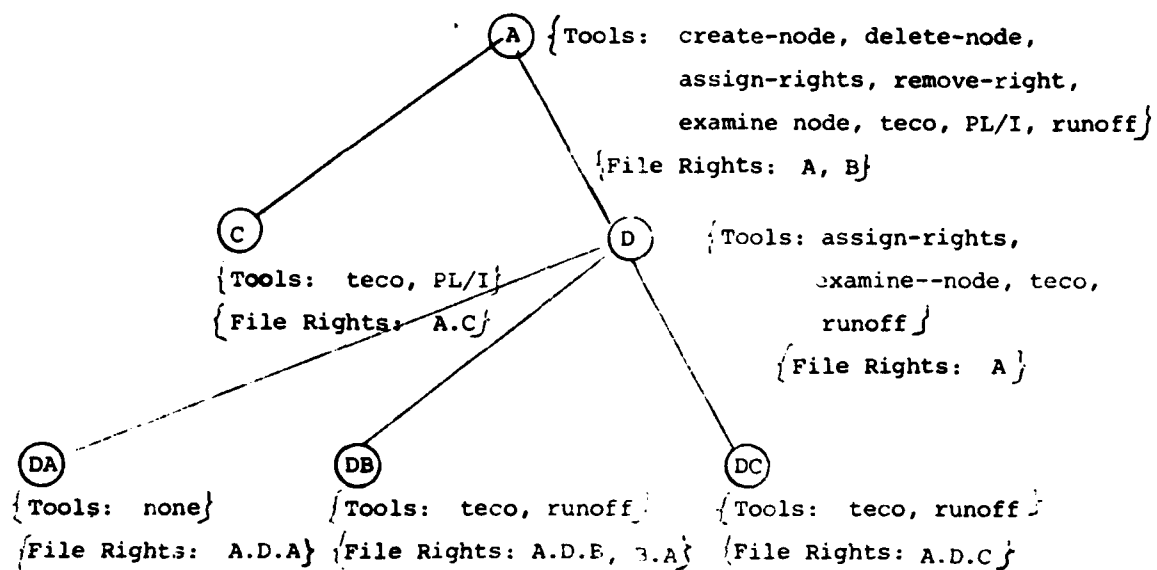
Node A Deletes Node B

Figure 6-4



Node A Revokes Node D's "Create-Node" Right

Figure 6-5



Node A Assigns File Right B.A to Node DB, But Not to Node D

Figure 6-6

### 6.1.2 The NSW File System

The NSW operating system provides its users with a uniform and coherent view of the NSW file system. The NSW file system is distributed across the set of network hosts participating in the NSW system configuration (i.e., those NSW hosts participating as File Bearing Hosts -- see Section 4.4.4); hosts participating in the NSW file system provide support for:

- Movement of files between "native" host file systems and NSW filespace
- Storage of, and access to files in NSW filespace
- Movement of files between NSW hosts (as necessary)
- Data translation during movement of a file from one NSW host to another (as necessary)

In the name of uniformity and coherence, NSW tries to make as many aspects of file access, movement and translation transparent to the NSW user.

The following example illustrates the file system functions listed above, including the level of transparency achieved by the NSW system. For the purposes of this example, an NSW file named "A" (containing text) is stored on the UCLA-CCN NSW host. An NSW user would like to create a copy of NSW file A in local directory <B> on the RADC-20 NSW host under the name "A.TXT". (The act of copying a file from NSW filespace into the "native" host file system is called "exporting a file from NSW filespace".) NSW provides a user command for "exporting" files, which requires only that the user designate the name of the NSW file to be exported, and where in the "native" file system it is to be placed (i.e., name, directory, password, etc.). To the NSW user, the file export is a single command, for which a positive or negative acknowledgement is received. However, to complete the export operation, the underlying NSW machinery must proceed as follows (see Figure 6-7):

1. Move file A from the NSW filespace on UCLA-CCN NSW host to the NSW filespace on the RADC-20 NSW host.
2. Translate the file as it is moved from UCLA-CCN to RADC-20 (file translation is required because UCLA-CCN and RADC-20 belong to two very different (i.e., incompatible) host families). As a minimum, character conversion (from EBCDIC to ASCII) is required.

3. Place a copy of the translated file in RADC-20 directory <B> under the name "A.IXT".
4. Notify the NSW user that the export operation has been successfully completed.

Note that all of the file "motion" and translation activity remains invisible to the NSW user.

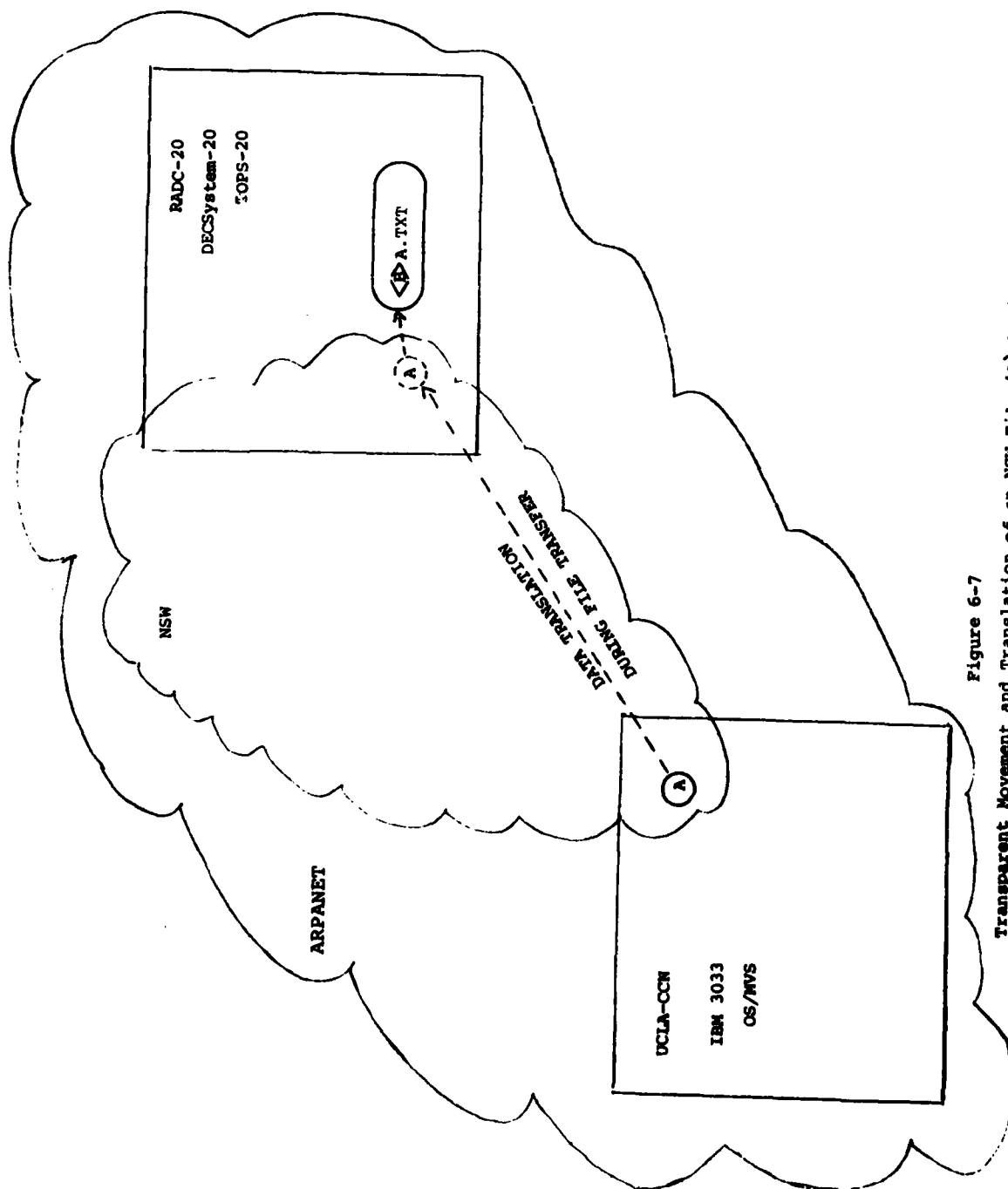


Figure 6-7  
Transparent Movement and Translation of an NSW File (A) During EXPORT

### 6.1.3 NSW Tools

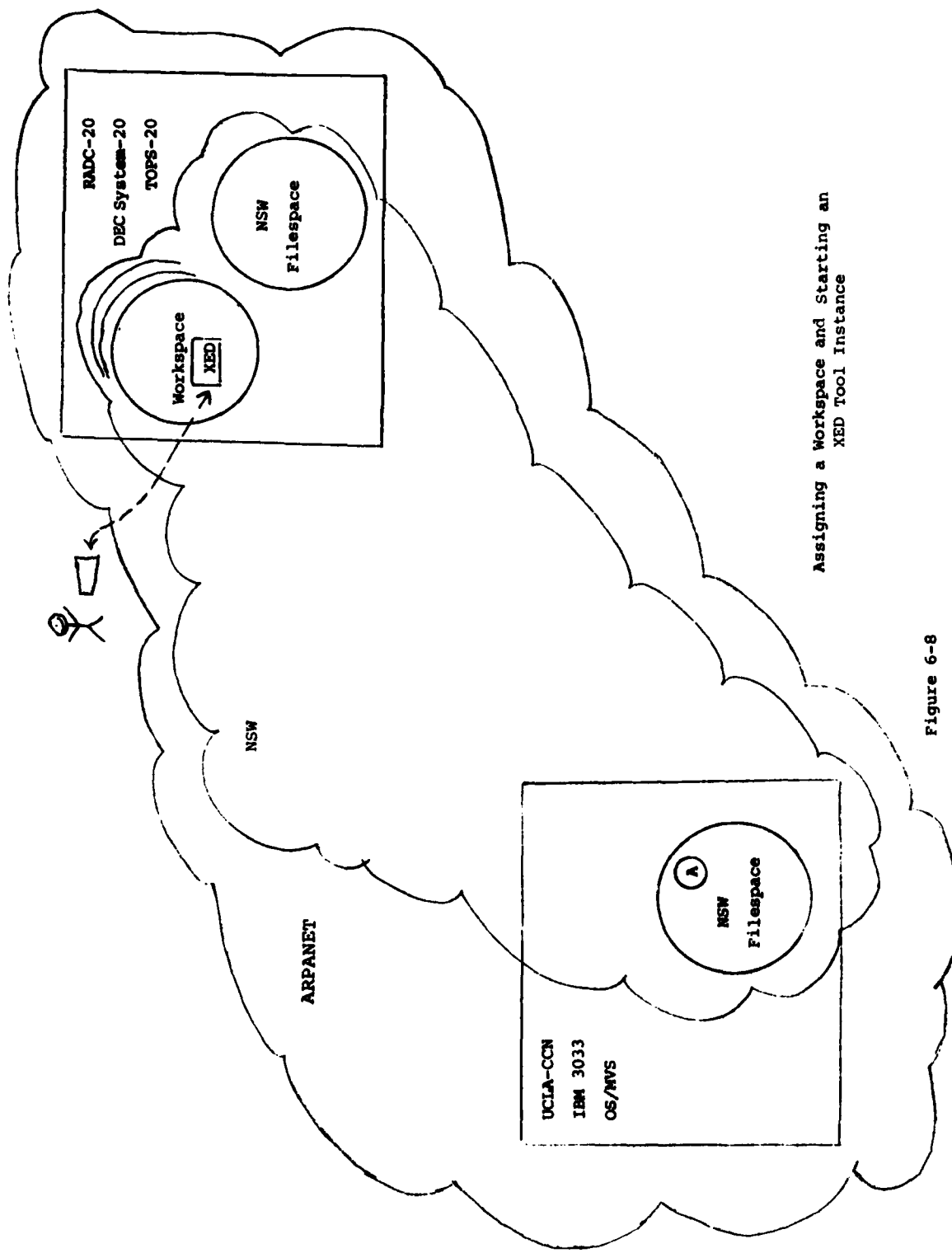
The NSW operating system provides support for four (4) different tool types; each of these may be characterized briefly as follows:

1. **Management Tools:** These tools are tightly coupled to and deeply imbedded in the NSW software; they are used to manipulate special NSW project management entities called "nodes" and "rights" (see Section 6.1.1) [28].
2. **NSW Tools:** These tools are implemented using special NSW operating system primitives specifically designed to support the construction of "NSW tools". No tools of this type presently exist. See the "NSW Tool Builder's Guide" [29] for a discussion of the NSW primitives available to tool builders.
3. **Batch Tools:** These tools are characterized by "staged" execution. That is, the user must supply all parameters, including input/output filenames, job control parameters, etc prior to tool execution; these tools are not user-interactive except during the period that runtime parameters are collected. A job number is assigned to each batch tool execution; the user may query the status of a job and automatically receives notification when the job has been completed (or terminated).
4. **Interactive Tools:** These tools are characterized by direct user contact and interaction during execution. They differ from Management and NSW tools in that they are neither imbedded in the NSW software nor make use of special NSW operating system primitives. Most interactive tools are copies of tools which have been built for and are regularly used in the "native" host (family) environment. Using a technique called "encapsulation", these "native" tools are regularly installed in the NSW system without modification. "Encapsulation" is the technique used to mediate all interactions between the tool and its execution environment (i.e., operating system calls); this includes (re)directing file references to either the NSW system or the "native" host operating system. Each interactive tool instance is executed in an assigned area called a "workspace"; the user model for interactive tools is complicated somewhat by the use of workspaces for tool execution (see the example below). Read operations cause NSW files to be moved

into the workspace for the duration of the tool session; new files or versions (of existing NSW files) are also placed in the workspace during tool execution. Upon termination of the tool session, the user may elect to "deliver" new and/or modified files from the workspace into NSW filespace.

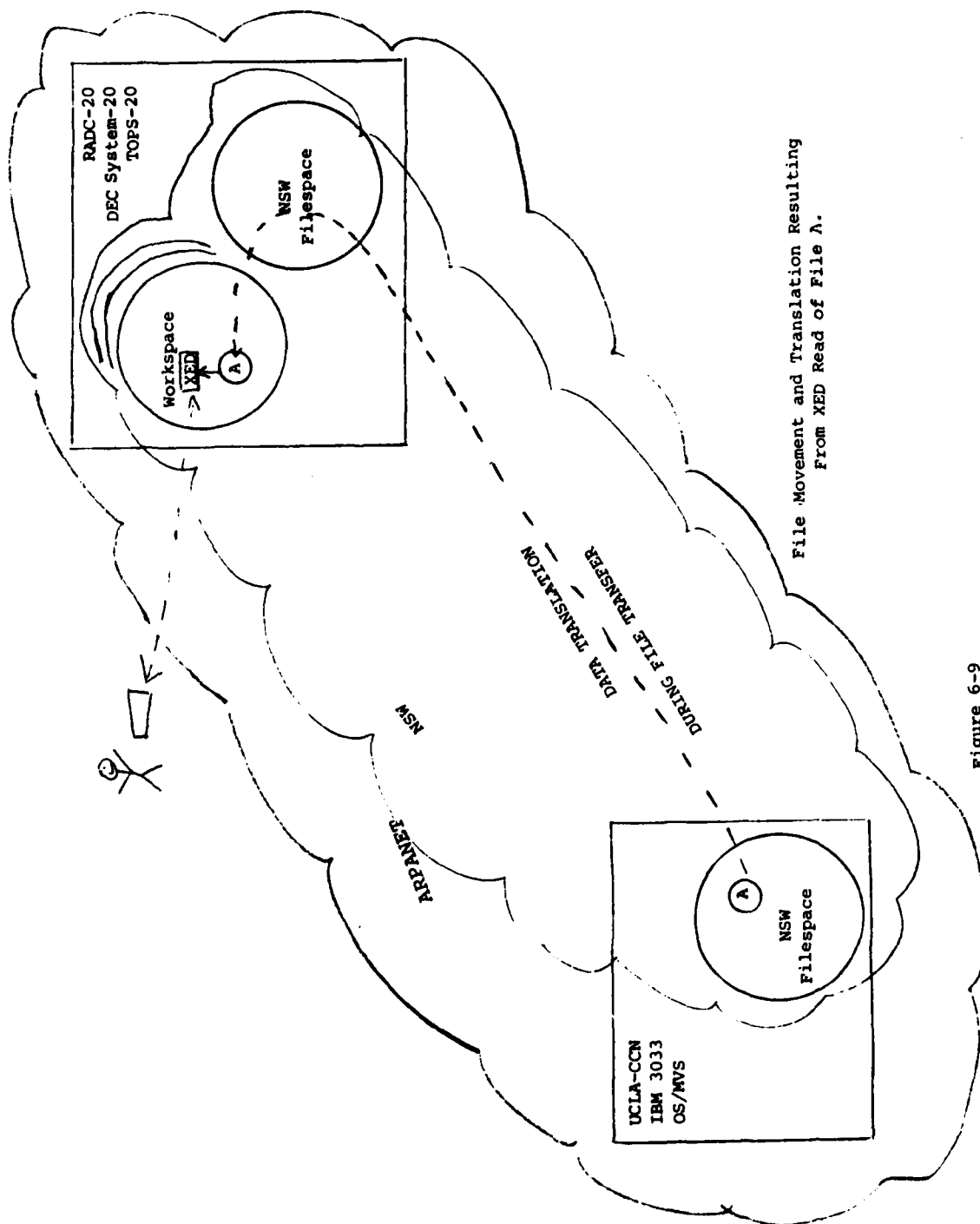
To illustrate, consider the manner in which the interactive editor XED (available on many TOPS-20 hosts, including the RADC-20) is used to modify NSW file "A" stored in NSW filespace at UCLA-CCN:

1. The NSW user invokes the XED tool.
2. NSW assigns a workspace for the XED tool instance, starts execution of XED in the assigned workspace, and places the NSW user in contact with the XED instance (Figure 6-8).
3. The NSW user reads file A from NSW filespace (Figure 6-9); it is translated and moved from UCLA-CCN to RADC-20, then into the assigned workspace, and accessed by XED.
4. The NSW user modifies file A, creating file B stored in the workspace (Figure 6-10).
5. The NSW user reads file A from the workspace, then modifies and creates a new version of file A (A' is used to distinguish between the original version of file A and the new workspace version) in the workspace (Figure 6-11).
6. The NSW user terminates his/her XED session (i.e., quits XED) and confirms delivery of the modified version of file A (A') and new file B into NSW filespace. Note: Because a new version of file A (A') was delivered into NSW filespace, file A is no longer stored at UCLA, but rather in the RADC-20 portion of NSW filespace where it was delivered (Figure 6-12).
7. NSW deallocates the workspace assigned to the XED instance.



Assigning a Workspace and Starting an  
XED Tool Instance

Figure 6-8



File Movement and Translation Resulting  
From XED Read of File A.

Figure 6-9

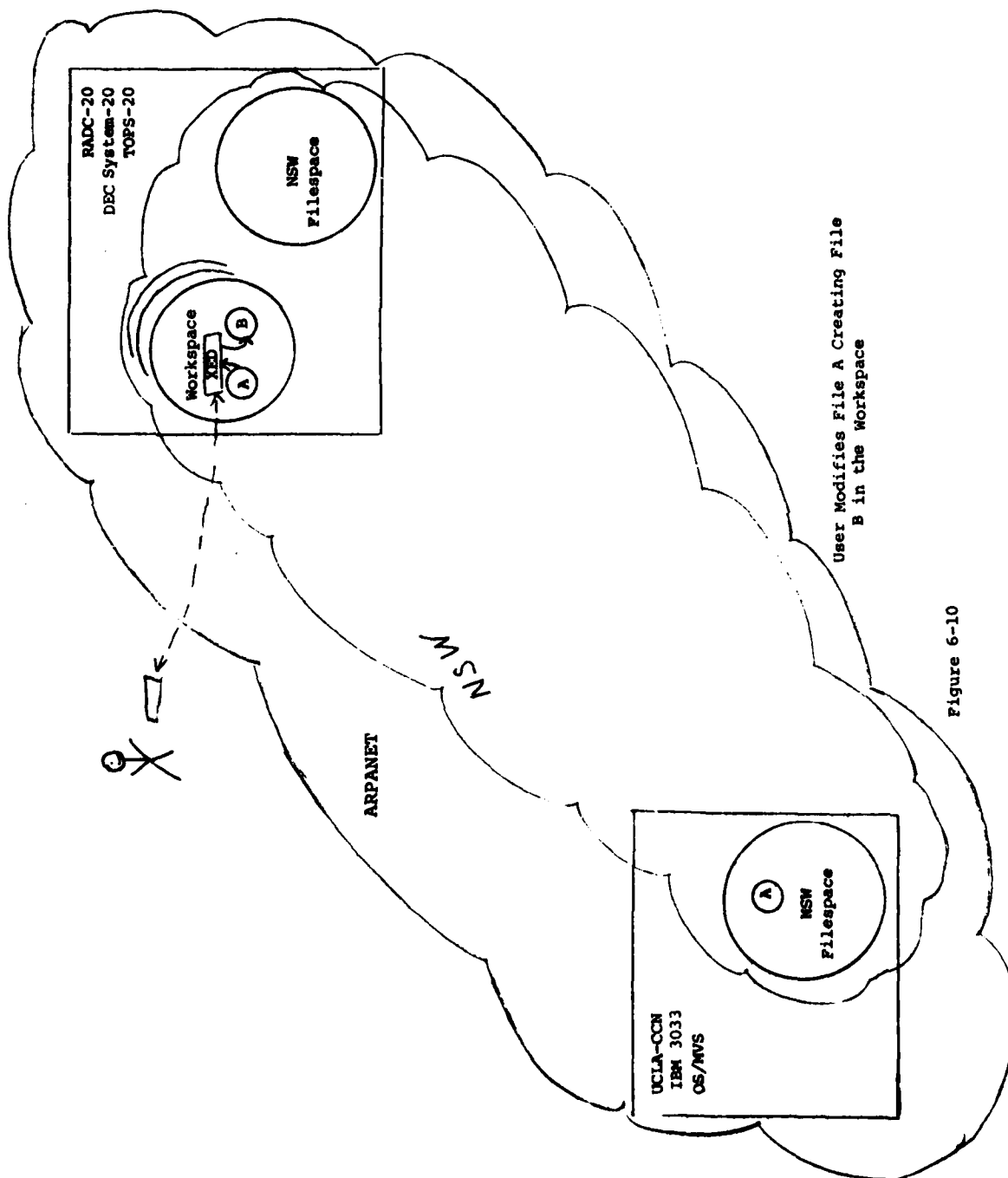
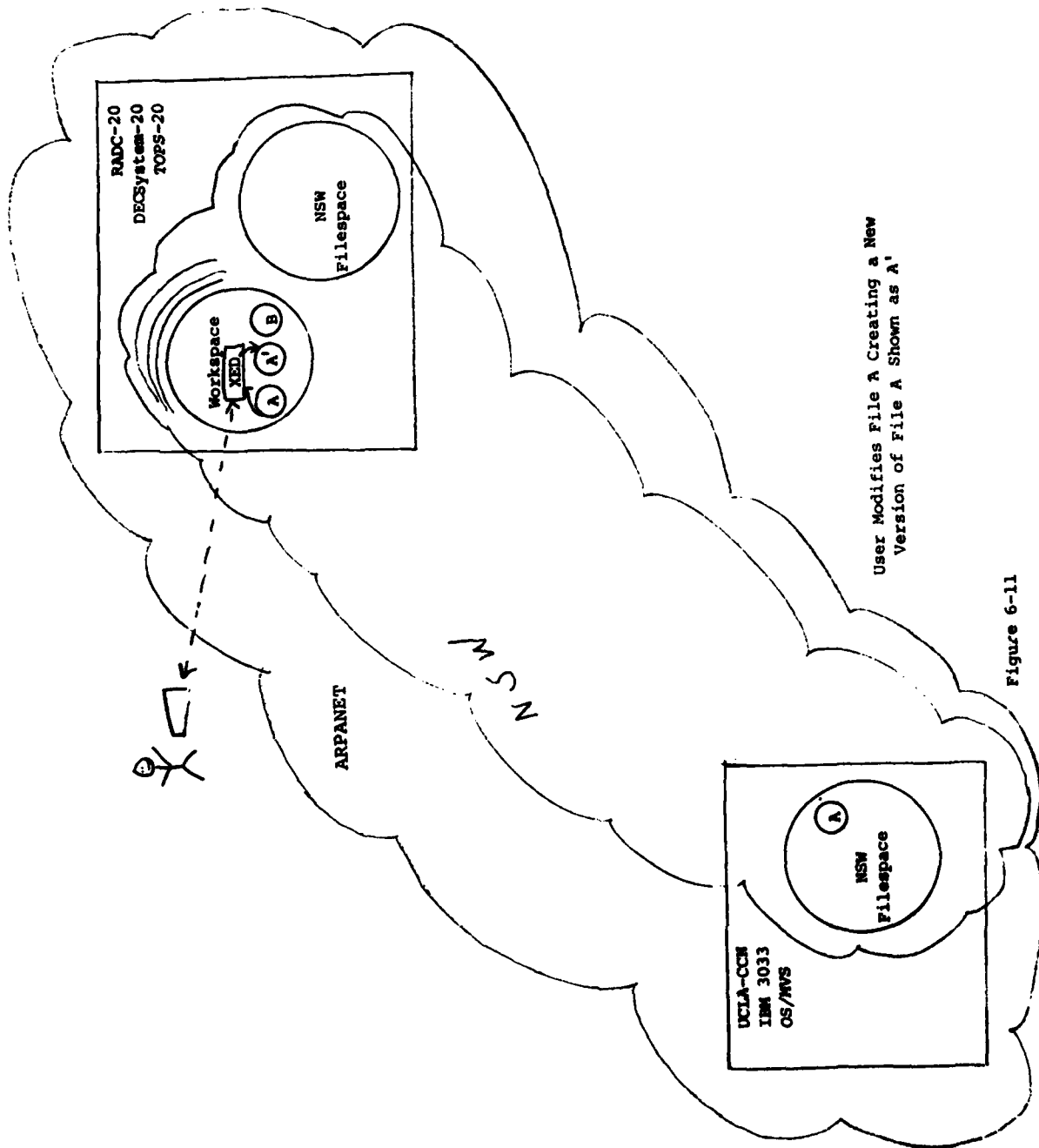
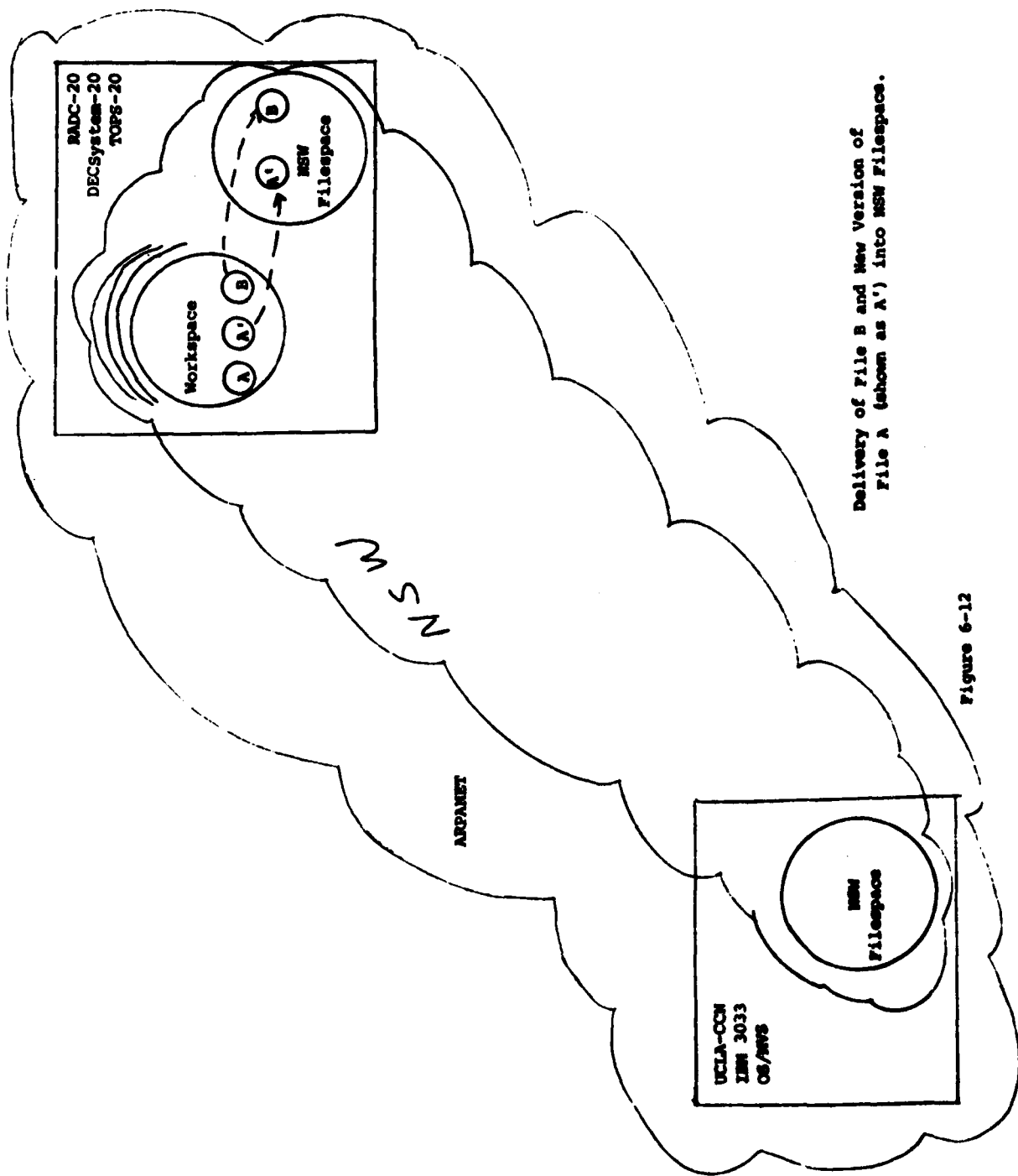


Figure 6-10



User Modifies File A Creating a New  
Version of File A Shown as A'

Figure 6-11



Delivery of File B and New Version of  
File A (shown as A') into MSW Filespace.

Figure 6-12

#### 6.1.4 NSW Command Language

NSW provides a number of user commands. These can be categorized and summarized as follows:

##### 1. User Session Management:

- Change password
- Terminate user session (logout)

##### 2. NSW File Manipulation

- Move file(s) between "native" host file system(s) and NSW filespace
- Copy, delete and rename NSW files
- Lock (set a semaphore on) an NSW file
- Change specificity (the "scope") of file key references

##### 3. Tool Session Management:

- Instantiate (request use of) a tool instance
- Abort an active tool session
- Resume an interrupted (e.g., due to host crash) tool session

##### 4. Information and Status Query:

- Inquire about the status of a batch job
- Display node attributes (e.g., sons and rights) of the user's login node
- List names of files (which may be accessed by the node)
- List file key "scopes" (see "NSW File Manipulation" commands above)

## **6.2 Current NSW User System Status (May 1981)**

The subsections which follow summarize the evolution, progress (both user-visible and user-invisible) and current status of the NSW system for the period covered by this report.

The NSW software system has improved dramatically in the last two years. Much of the progress is of a form which is not directly user-visible (primarily because so many NSW services are transparent to users). At the same time, the progress is so substantial that the absence of this progress would mean an NSW system for which NSW users would most definitely discern ("feel") a difference. During the contract period covered by this report, the NSW system has:

1. Become easier to operate
2. Embodied greater economy of concept (and has become easier to use)
3. Reduced resource consumption dramatically
4. Improved in performance and user responsiveness
5. Achieved more reliable and robust operation

### 6.2.1 NSW Command Language Status

The NSW user interface (including the NSW command language and its interpretation) is largely implemented by the Front End component (and its interactions with other NSW components, such as the Works Manager). Perhaps the most significant achievement in this area has been the addition of a new (alternative) Unix-based Front End component. The Unix Front End implements a somewhat richer user interface and command set (including, for example, "immediate command return mode" which allows NSW users to initiate multiple, concurrently executing NSW commands -- see Section 6.1.4). However, the syntax for NSW commands implemented by the Unix Front End differs visibly from that which was adopted for the TOPS-20 Front End. Plans have been made for installing the Unix Front End on the AFLC "base system" configurations (PDP-11's running Unix) for the AFLC Technology Demonstration. This new component has not yet been released for O/A or installed in the NSW User System configuration.

No new commands or user capabilities of the kind mentioned above (e.g., "immediate command return mode") have been incorporated into the TOPS-20 Front End implementation. The command language implemented corresponds to the commands listed in Section 6.1.4. To provide the NSW user with the best user interface possible, significant TOPS-20 Front End modifications were made. These modifications addressed the following user concerns:

- Component reliability,
- Tool connection management,
- Error reporting, and
- Consolidation of user interface (including control character functionality, consistent prompting, command consolidation, etc.).

### 6.2.2 NSW File System

Improvements in the NSW file system largely depend upon the evolution of the various host family implementations of the File Package component. Progress in this area can be summarized as follows:

1. A new, more efficient implementation of the TOPS-20 File Package component was introduced.
2. The IBM and Multics File Package implementations became more dependable and reliable.
3. The following new file types (by host family) were incorporated into the corresponding host family implementations of the File Package:

#### TOPS-20 (2):

- 10X-TEXT
- 10X-SOSTXT

#### IBM ( ):

- 360-ASM-OBJ
- 360-MACR20-OBJ
- 360-COBOL-SEQ-SOURCE
- 360-ASM-SOURCE
- 360-PLM80-OBJ
- 360-PLM80-SOURCE
- 360-OVERPRINT
- 360-ASM80-SOURCE

#### Multics (2):

- MTX-TEXT
- MTX-LIBRARY

Unix:

**\*\* none \*\***

- (Note: A Unix host family implementation of the File Package component does not presently exist.)

4. NSW file transformations have been comprehensively tested and major problem areas identified
5. Protocol modifications have been designed to improve user reliability, user error reporting, and performance

Presently, the best (if not the only reasonable) measure of NSW file system status is the number of file types supported by each host family (see Table 6-1).

- TOPS-20 (3):
  - 10X-BINARY
  - 10X-TEXT
  - 10X-SOSTEXT
- Multics (2):
  - MTX-TEXT
  - MTX-LIBRARY
- IBM (30):
  - 360-TEXT
  - 360-ORIGINAL-BIN
  - 360-MACRO20-SOURCE
  - 360-ASM-OBJ
  - 360-BINARY
  - 360-MACRO20-OBJ
  - 360-OBJECT
  - 360-COBOL-SEQ-SOURCE
  - 360-COBOL-SOURCE
  - 360-ASM-SOURCE
  - 360-PLI-CC-SOURCE
  - 360-PLI-CC-CARDS

Table 6-1

NSM File Types by Host Family

Table 6-1 (continued)

- 360-PLI-CARDS
- 360-JCL
- 360-ORIGINAL
- 360-PLM80-OBJ
- 360-CMS2M-OBJ
- 360-OVERPRINT
- 360-SPPCOBOL-SOURCE
- 360-PLM80-SOURCE
- 360-PLI-SOURCE
- 360-CMS2M-SOURCE
- 360-ASM80-SOURCE
- 360-LOAD
- 360-FORTRAN-SOURCE
- 360-CARDS
- 360-LIST
- 360-PRINT
- 360-KEYPUNCH

Unix:

\*\* not applicable \*\*

Table 6-1

NSM File Types by Host Family

### 6.2.3 NSW Tools

Significant progress has been achieved in establishing tool management procedures and improving the level of tool support provided by the NSW system.

The NSW Tool Manager has been largely responsible for procedural advancements. The Tool Manager has prepared a "Tool Quality Management and Control Plan" [30] and a "Tool Installation Guide" [31], which together address the following tool-related issues:

- [Tool] selection
- Acceptance
- Installation and maintenance
- Configuration management
- New tool development
- Administration and legal problems

Improvements in the level of tool support provided by the NSW system are much like file system improvements (see Section 6.2.2) largely determined by evolution of the various host family implementations of the Foreman and Batch Job Package components; progress in this area is summarized below by host family:

#### TOPS-20 (Foreman)

- Support for allocation and archiving of tool sessions
- Improved tool session management, including user prompts for workspace file access
- Simplification of operator interface
- Improved reliability mechanisms
- Proposed protocol modifications to improve performance

#### Multics (Foreman)

- More reliable and resilient implementation
- Support for aborting and rerunning tool sessions
- Support required for the NSW reliability scenarios
- Improved user interface and workspace management
- Support for program execution

IBM (Foreman, and Batch Job Package)

- Improved operator support for batch jobs
- Full interactive tool support (excluding NSW reliability scenarios)

A measure of tool support status (and progress as well) may be gleaned by:

- Identifying the new tools installed in the NSW system during the contract period covered by this report (Table 6-2)
- Summarizing the number of tools installed on each host by "NSW" tool type as defined in Section 6.1.3 (Table 6-3)
- Itemizing NSW tools by name and User System Host (Table 6-4)(32)
- Itemizing NSW tools by "generic" tool type and User System host (Table 6-5)(33)

TOPS-20 (5):

CONCORDANCE  
ECL  
FMTBCPL  
PSAVE  
SRCCOM

IBM (24):

ASMB0  
ASMCOMP  
ASMLINK  
COBOL  
COMPRESS  
CREATEC  
CREATEL  
CREATEU  
CREATEP  
DISPLAY  
FORTCOMP  
FORTLINK  
GETC  
GETO  
GETP  
LIBMAINT  
MERGELIB  
PLICOMP  
PLILINK  
PLM80  
PUTC  
PUTO  
PUTP  
TSOEDIT

MULTICS (5):

ADA  
ADA-LSTAT  
PDL  
RUN  
RUN-ADA

Table 6-2

New NSw Tool by Host Family

	Management	NSW	Batch	Interactive	Total
ISIE	5	0	0	20	25
ISIC	0	0	0	21	21
RADC-20	0	0	0	16	16
CCN	0	0	30	2	32
RADC- MULTICS	0 ----	0 ----	0 ----	13 ----	13 ----
TOTAL	5	0	30	72	107

Table 6-3

Number of NSW Tools by User System Host and "NSW" Tool Type

GENERIC NAME	- - - - HOST SPECIFIC SUFFIX				
	USC- ISIE	USC- ISIC	RADC- 20	RADC- MULTICS	UCLA- CCN
ADA				-RM	
ADA-LSTAT (library-status)				-RM	
ALM				-RM	
ASM80					-UC *
ASMCOMP					-UC *
ASMLINK					-UC *
ASSIGNRIGHTS (none)					
BASIC				-RM	
BCPL	-IE	-IC	-R2		
BDDT	-IE	-IC	-R2		
CMS2M					-UC *
COBOL					-UC *
COMPRESS					-UC *
CONCORDANCE	-IE				
CREATEC					-UC *
CREATEL					-UC *
CREATENODE (none)					
CREATED					-UC *
CREATEP					-UC *
DELETENODE (none)					
DESCRIBE	-IE	-IC	-R2		
DISPLAY					-UC
ECL	-IE				
EMLOAD		-IC			
EXAMINENODE (none)					
FMTBCPL	-IE				
FORTCOMP					-UC *
FORTLINK					-UC *
FORTIRAN				-RM	-UC *
FTP	-IE	-IC	-R2		
GETC					-UC *

Table 6-4

NSW Tools by Name and User System Host  
 (Note: Batch tools are marked with an asterisk (\*\*))

Table 6-4 (continued):

GENERIC  
NAME

- - - -

HOST SPECIFIC  
SUFFIX

- - - - -

	USC- ISIE	USC- ISIC	RADC- 20	RADC- MULTICS	UCLA- CCN
GETO					-UC *
GETP					-UC *
HELP				-RM	
HOSTAT	-IE	-IC	-R2		
LODT	-IE	-IC	-R2		
JIGSAW	-IE	-IC	-R2		
LIBMAINT					-UC
LINKER	-IE	-IC	-R2		
MACRO	-IE	-IC	-R2		
MACRO20		-IC			-UC *
MERGE LIB					-UC *
MRUNOFF	-IE	-IC	-R2		
NETSTAT	-IE	-IC	-R2		
POL				-RM	
PL1				-RM	
PLI					-UC *
PLIBGO					-UC *
PLICOMP					-UC *
PLILINK					-UC *
PLM80					-UC *
PRIM		-IC			
PSAVE	-IE				
PUTC					-UC *
PUTO					-UC *
PUTP					-UC *
QEDX				-RM	
REMOVERIGHTS	(none)				

Table 6-4

NSW Tools by Name and User System Host

[Note: Batch tools are marked with an asterisk (\*\*\*)]

Table 6-4 (continued):

GENERIC NAME	- - - -	HOST SPECIFIC SUFFIX	- - - - -
RUN			-RM
RUN-ADA			-RM
RUNOFF			-RM
SMITE			-RM
SOS	-IE	-IC -R2	
SPELL	-IE	-IC -R2	-RM
SPPCOBOL			-UC *
SRCCOM	-IE	-IC -R2	
TECO	-IE	-IC -R2	
TSUEDIT			-UC
UYK20		-IC	
XED	-IE	-IC -R2	

Table 6-4

NSW Tools by Name and User System Host

[Note: Batch tools are marked with an asterisk ("\*")]

TYPE OF TOOL	GENERIC NAME	HOST SPECIFIC SUFFIX				
		USC- ISIE	USC- ISIC	RADC- 20	RADC- MULTICS	UCLA- CCN
ASSEMBLER:	ALM				-RM	
	ASMCOMP					-UC *
	MACRO	-IE	-IC	-R2		
COMPILER:	ADA				-RM	
	BASIC				-RM	
	BCPL	-IE	-IC	-R2		
	COBOL					-UC *
	FORTCOMP					-UC *
	FORTTRAN				-RM	-UC *
	PL1				-RM	
	PLI					-UC *
	PLIBGO					-UC *
	PLICOMP					-UC *
	SMITE				-RM	
	SPPCOBOL					-UC *
CROSS ASSEMBLER:	ASM80					-UC *
	MACRO20		-IC			-UC *
CROSS COMPILER:	CMS2M					-UC *
	PLM80					-UC *
DEBUGGER:	BDDT	-IE	-IC	-R2		
	IDDT	-IE	-IC	-R2		

Table 6-5

NSW Tools by Generic Tool Type and User System Host  
 [Note: Batch tools are marked with an asterisk (\*\*)]

Table 6-5 (continued):					
TYPE OF TOOL	GENERIC NAME	-	-	HOST SPECIFIC SUFFIX	-
EDITOR:					
	QEDX			-RM	
	SOS	-IE	-IC	-R2	
	TECO	-IE	-IC	-R2	
	TSOEDIT				-UC
	XED	-IE	-IC	-R2	
EXECUTION SUPPORT					
	RUN			-RM	
	RUN-ADA			-RM	
FILE UTILITY:					
	DISPLAY				-UC
	FTP	-IE	-IC	-R2	
	SRCCOM	-IE	-IC	-R2	
INTERPRETER:					
	ECL	-IE			
	PLIBGO				-UC *
LIBRARY UTILITY:					
	ADA-LSTAT			-RM	
	COMPRESS				-UC *
	CREATEC				-UC *
	CREATEL				-UC *
	CREATEO				-UC *
	CREATEP				-UC *
	GETC				-UC *
	GETO				-UC *
	GETP				-UC *
	LIBMAINT				-UC
	MERSELIB				-UC *
	PUTC				-UC *
	PUTO				-UC *
	PUTP				-UC *

Table 6-5  
NSW Tools by Generic Tool Type and User System Host  
(Note: Batch Tools are marked with an asterisk ("\*"))

Table 6-5 (continued):					
TYPE OF TOOL	GENERIC NAME	-	-	HOST SPECIFIC SUFFIX	-
LINKER:					
	ASMLINK				-UC *
	FORTLINK				-UC *
	LINKER	-IE	-IC	-R2	
	PLILINK				-UC *
NSW MANAGEMENT:					
	ASSIGNRIGHTS	(none)			
	CREATENODE	(none)			
	DELETENODE	(none)			
	EXAMINENODE	(none)			
	REMOVERIGHTS	(none)			
PRIM EMULATOR:					
	EMLOAD		-IC		
	JIGSAW	-IE	-IC	-R2	
	PRIM		-IC		
	U1050		-IC		
	UYK20		-IC		
PROGRAMMING AID:					
	CONCORDANCE	-IE			
	FMTBCPL	-IE			
	PDL			-RM	
	PSAVE	-IE			
SPELLING CORRECTOR:					
	SPELL	-IE	-IC	-R2	-RM

Table 6-5

NSW Tools by "Generic" Tool Type and User System Host

Table 6-5 (continued)

TYPE OF TOOL	GENERIC NAME	-	-	-	HOST SPECIFIC SUFFIX	-	-	-
SYSTEM INFO:								
	DESCRIBE	-IE		-IC	-R2			
	HELP					-RM		
	HOSTAT	-IE		-IC	-R2			
	NETSTAT	-IE		-IC	-R2			
TEXT PROCESSOR:								
	MRUNOFF	-IE		-IC	-R2			
	RUNOFF					-RM		

Table 6-5

NSW Tools by "Generic" Tool Type and User System Host  
 [Note: Batch tools are marked with an asterisk ("\*")]

#### 6.2.4 User Documentation

At different times in the project's history, various NSW user oriented documents have been prepared. These include:

- "The NSW Users' Reference Manual" [35]
- "The NSW Users' Guide" [36],
- "Interim NSW Managers' Tools" [37]
- "The NSW Beginners' Guide" [38]

Currently, the only document that is maintained updated is the "NSW User's Reference Manual". GSG updates this document for each new system release. GSG was also tasked with preparation of an "NSW User Impact Bulletin" for each NSW system release.

The "NSW User Impact Bulletin" [38] augments the "NSW Users' Reference Manual". The Bulletin was designed to be a highly modular compendium of notes:

- identifying all user visible changes associated with a new system release. These changes can take many forms, including new or changed command functionality, elimination of bugs found in earlier releases, improved error messages, and new tools, implementation or host facilities.
- identifying all known user issues and limitations that exist in the current release, and suggesting how to respond to each.
- augmenting existing NSW user documents as needed. For example, the UIB attempts to clarify, expand, and provide additional background on user features that may be misunderstood or are counter-intuitive in their application.
- supplementing the NSW user documentation by supplying information not otherwise available. For example, the UIB comprises the only uniform documentation of operational differences between tools in the NSW environment and their behavior in the native host environment (with the possible exception of UCLA tool documentation -- see below).

Items which fall into the above categories are typically presented as incremental updates relative to the last NSW system release. Such items are generally removed from the "User Impact

Bulletin" within the next two NSW system releases. However, some items remain in the "User Impact Bulletin" until they are incorporated into more suitable user documentation (e.g., the "NSW Users' Reference Manual"). With the above requirements in mind, the "User Impact Bulletin" was designed so that incremental modifications could be made easily and quickly.

"User Impact Bulletins" have been prepared and distributed for NSW releases 4.1 and 5.0 [39,40]. In addition, a "special" version of the "User Impact Bulletin" for NSW 4.1 [41], containing only those items to be considered as candidates for inclusion in the "NSW Users' Reference Manual", was prepared for ACC. Preparation and distribution of updates to the NSW 4.1 "User Impact Bulletin" [42] was testimony to viability of the plan for quick, incremental modification of the "User Impact Bulletin". After the "User Impact Bulletin" for NSW 4.1 had been completed, a plan for maintaining the "User Impact Bulletin" was prepared and distributed to interested parties [43].

During development of the "User Impact Bulletin", GSG conducted a review of the existing NSW user documentation [44]. Based in part on this review, user documentation requirements were identified and documented. These requirements were then compared to the current state of NSW user documentation, and specific tasks were proposed to address known deficiencies [45].

In the separate area of NSW tool documentation (describing operability of individual tools which have been installed in the NSW environment), UCLA has provided documentation for the following IBM tools installed in NSW.

- "Using The UCLA Native Language Processing Tool Kits" [46]
- "Using The DISPLAY Tool" [47]
- "Using The TSOEDIT Tool" [48]
- "Using the UCLA PL/I Tool Kit" [49]
- "Using The UCLA Interim Library Management Tool Kit" [50]

#### 6.2.5 Future Directions and Problem Areas

Probably the most important development planning that occurred during the contract period covered by this report has been:

- Formulation of the functional and performance attributes required for the AFLC Technology Demonstration,
- Recognition of NSW structural, performance and feature deficiencies relative to Demonstration needs, and
- Preparation of a development plan for addressing Technology Demonstration support.

During the latter half of the contract period, an NSW Analysis Group (chaired by Bob Thomas of BBN) was created. This internal group was assigned the following tasks:

- Identify NSW requirements (for the AFLC Technology Demonstration)
- Prepare a functional specification which conforms to the identified requirements
- Develop an evolution plan for realizing the specified NSW functions

Based on input from NSW users and contractors, the Analysis Group identified and documented NSW requirements in a document referred to as the "NSW Concept Paper" [51]. GSG also proposed requirements [52] and reviewed the requirements distributed by the NSW Analysis Group [53]. Subsequently, the Analysis Group prepared a revised NSW Functional Specification [54]. This plan will serve as a guideline for NSW development activities during the Technology Demonstration period. A summary of NSW system areas likely to receive attention, and examples of proposed improvements follow:

- User Interface
  - . User I/O commands (e.g., display contents of a file on the user's terminal)
  - . User information and status reporting commands (e.g., system status, file attributes, etc.)

- Improved error reporting (e.g., report abnormal conditions to the NSW Fault Logger)
- Simplified User Model
  - Single, uniform namespace for files and tools (executable programs)
- Tool Support
  - Tool chaining (e.g., compile, then link, then load and execute by invoking one rather than three tools)
  - User "owned" workspaces
- File System
  - User "owned" segments of NSW filespace
  - Improved file translation and typing
  - Additional file access methods (e.g., direct access)
  - Reliable transfer of long files
- Performance
  - Optimized file motion and access protocols
  - Support for "native" services (i.e., tools and servers which operate in port "outside" of NSW, in the "native" host environment)

#### 6.2.6 NSW Availability

NSW System Operations (NSWOPS) is responsible for operating the NSW User System configuration on behalf of the NSW user community. A goal of GSG's NSW operations group has been to achieve maximum NSW availability during operator attended periods (9 AM - 5 PM EST, Monday - Friday). This means that NSW system downtime should correlate as closely as possible to the duration of NSW host outages (especially the "Core System" Host -- see Section 4.4.3).

To provide a measure of service provided by NSWOPS to the NSW user community, aggregate statistics of NSW availability for the period October 8, 1979 through November 10, 1981 are summarized below:

	Time Period	
	Operator Attended only  (M-F, 9-5) -----	Operator Attended and Unattended  (M-F, All Day) -----
Average Weekly Downtime (hours & minutes)	3:38	44:02
NSW Availability (%)	91%	74%

The above figures are based on "Core System" availability statistics accumulated since October of 1979. (Note: NSW can continue to operate in the face of some non-"Core System" host outages; however, the NSW system remains completely unavailable to users during "Core System" Host outages.)

## 7.0 NSW System Operations

During 1978, an NSW Operations Office (NSWOPS) was established at the Rome Air Development Center, Griffiss Air Force Base, Rome, New York. Since that time, on-site GSG staff members have been expanding and evolving the NSW Operations Center which is responsible for the NSW User System. NSWOPS tasks and responsibilities are reviewed in the first section which follows. The second section discusses NSWOPS activities and contributions during the contract report covered by this period.

### **7.1 NSWOPS Responsibilities**

The major responsibilities of the NSW Operations are summarized below:

1. Install and operate NSW system releases in the NSW User System configuration
2. Devise and automate operational procedures; apply these (automated) procedures to the NSW User and Candidate System configurations.
3. Provide technical support to the NSW user community, assuring expeditious resolution of all user questions and problems; develop an accounting system for tracking all user inquiries and Software Trouble Reports (STR's).
4. Provide user information services including an NSW HELP facility and usage reporting for the NSW operations database.
5. Prepare NSW user documentation including a "User Impact Bulletin" summarizing pertinent information for use of the NSW system and its tools, including: improvements, outstanding bugs, limitations, anomalies and procedures for dealing with and/or circumventing these.
6. Implement configuration control and auditing procedures in the NSW User System configuration. Maintain the NSW Documentation Repository, placing NSW documentation under configuration control
7. Support PDC Quality Assurance (Q/A) activities by operating the NSW Candidate System and participating in Q/A testing under PDC's direction.

A more detailed discussion of NSWOPS activities and achievements may be found in the Section (7.2) which follows.

## 7.2 NSWOPS Activities and Accomplishments

The major accomplishments and contributions of the NSW Operations Organization (NSWOPS), in each major area of responsibility (see Section 7.1) are summarized below:

### 1. User System Operations

This task primarily concerns the application of (automated) operational procedures (see 2 below) to operation of the NSW User System configuration.

Due to autonomous operational requirements of the IBM and Multics hosts, the scope of NSWOPS' operational activities is limited to the set of TOPS-20 ARPANET hosts participating in the User System configuration. The IBM and Multics hosts operate primarily in an "autostart" mode; little if any operator intervention is ever required (except, possibly to track-down bugs).

The set of participating TOPS-20 hosts must, of necessity, include the "Core System" Host. Most of the operational requirements imposed by the NSW system are focused on this "Core System" host, where control and synchronization are centralized and the critical system databases (user, file and tool catalogs) are maintained.

NSWOPS strives to achieve maximum system availability for the NSW user community. NSWOPS has been very successful in meeting this self-imposed goal (see Section 6.2.6 for details).

### 2. NSW Operations Procedures

NSWOPS is tasked with development of manual and automated procedures for operating NSW system configurations. Due to the nature of the operator interface, procedures for operating NSW systems tend to be many, varied and complex. In spite of this, much has been achieved during the contract period in the areas of procedure development and automation.

NSW operators must, minimally, be concerned with procedures for:

- Installing new NSW system releases, and
- Operating installed releases
- . Development and documentation of a "kernel" set of operations procedures for release installation and operation, a significant undertaking, was completed during the contract period covered by this report [55]. In addition, the following procedures were partially or completely automated:
  - Installation of new NSW system releases
  - Recognition of:
    - NSW failures
    - NSW ARPANET host outages
    - Restoration of host service
  - Operator notification of host failures and restarts
  - Host clean-up and restart

### 3. User Support and Technical Assistance

NSWOPS is logically (and has officially been designated) the focal point for resolution of user questions and problems. In support of this responsibility, NSWOPS was tasked with the development of a Software Trouble Report accounting system which could be used to track outstanding user questions, bug reports and new feature requests (as well as contractor-reported bugs and deficiencies). GSG designed and implemented a highly parametric and flexible accounting system based on the existing ARPANET mail facility. However, after:

- Considerable experience had been gained with the mail-based system, and
- Volume had increased dramatically, and
- Requirements for controlled interaction between NSW users became apparent,
- . A protocol-oriented tool for MONitoring STR's (MONSTR) was conceived, designed and implemented by Massachusetts Computer Associates (COMPASS). For more on STR processing, see Section 5.5.

#### 4. User Information Services

NSWOPS serves as a conscientious, accurate source of information about NSW system features status and use. NSWOPS has been responsible for developing a number of vehicles for communicating information to NSW users:

- Periodic usage reporting
  - Automated status reporting
  - An integrated, user-oriented information utility (HELP facility)
  - The "NSW User Impact Bulletin" (see 5 below or Section 6.2.4)
- At the beginning of each week, NSWOPS prepares, publishes and distributes a report detailing availability and utilization of the NSW User System for the preceeding week. A sample "Weekly Usage Report" appears in Appendix D.

System status reporting is an integral part of the automated host failure and restart recognition process (see 2 above). Users may query current system status information (maintained by this automated operator utility) through a special TOPS-20 command at RADC-20 or the NSW HELP facility (see below).

NSWOPS was tasked with development of an NSW HELP facility for the NSW user community. First, a specification detailing:

- Information requirements
  - Functional requirements
  - User interface
- was prepared for review [56]. At this stage it became apparent that ambitious requirements could be realized through a facility, of great generality and power, which could also be constructed at a very reasonable cost. The product of this effort is a high-quality, integrated information utility which:
    - Provides a uniform interface to wide variety of information

- Allows information to be added or exchanged without modifying software
  - Provides an easily extended topical index, and
  - Is user-oriented and friendly
- This facility, available as an NSW tool, provides NSW users with on-line, interactive access to the following information:
- "NSW User Impact Bulletin"
  - "NSW Users' Reference Manual"
  - "Interim NSW Managers' Tools"
  - "NSW Documentation Summary"
  - DESCRIBE Tool databases
  - HELP-RM tool databases
  - Current NSW system status (as described above)
- In addition, the HELP Facility provides a netmail conduit between NSW users and NSWOPS. It has been provided so that NSW users may send written communications to NSWOPS from within NSW. Among other things, this provides NSW users with a mechanism, internal to NSW, for initiating inquiries and Software Trouble Reports (see 3 above). We refer interested readers to Appendix C which provides a thorough overview of the NSW HELP facility -- its goals, functions and information.

## 5. User Documentation

The "User Impact Bulletin" is NSWOPS' major responsibility in the area of user documentation. In addition, NSWOPS has shown a conscientious interest in understanding and responding to user documentation needs. We refer the reader to Section 6.2.4 where both of these items are discussed in detail.

## 6. Configuration Management

Configuration management is an integral part of managing the NSW software development process (see Section 5.6). NSWOPS is responsible for implementing configuration management procedures devised by PDC in the NSW User System configuration. NSWOPS controls

changes to (i.e., the introduction of new configuration items into) the NSW User System configuration and performs periodic audits to verify conformance with the baseline configuration (this is most important for the IBM and Multics host configurations which are not directly under NSWOPS' operational control). In addition, NSWOPS is responsible for controlling and auditing changes to the NSW Documentation Repository, which is the focal point for collection and dissemination of all NSW documentation. Procedures for maintaining the NSW Documentation Repository have been devised and partially automated [57]. These procedures automatically:

- Update an index of documentation repository contents, and
  - Generate a formatted "NSW Documentation Summary" which is publically available to interested NSW contractors and/or users
- The Documentation Repository index is in a format compatible with and accessible through the NSW HELP facility (see 4 above).

#### 7. Q/A Testing Support

NSWOPS supports PDC's Q/A testing activities in a number of ways:

- Operation of the NSW Candidate System during Q/A assessment periods
  - Assistance in preparation and/or modification of test plans, methodologies and scripts
  - Application of tests to new NSW system releases
  - Assistance in preparation of test scenarios and reports
- Q/A assessment activities are very much a joint effort between PDC and NSWOPS. It is, therefore, not easy to distinguish the specific responsibilities (other than Candidate System operation) of one organizational entity from the other. We refer the reader to Section 5.4 for a more detailed discussion of GSG's (PDC and NSWOPS) Quality Assurance activities.

## 8.0 AFLC Technology Demonstration

GSG (PDC) has played a major role in the planning and development of the NSW AFLC Technology Demonstration. PDC has contributed directly to the demonstration planning activities through participation in the "NSW Working Group" created for that purpose. At RADC's request, technology transfer [58] and demonstration milestone [59] plans were proposed for the demonstration.

Early in the planning stages, the NSW working Group concentrated on identifying and developing application scenarios for the technology demonstration. GSG proposed a procedure for the identification and development of these application scenarios [60].

The following four application areas were identified by the participating AFLC sites as candidate demonstration scenarios:

1. Configuration Management
2. Emulation Support
3. NSW Tool Repository
4. ADA Training

Configuration Management was selected by the AFLC's as the demonstration scenario which should receive highest priority. GSG was assigned responsibility for coordinating and developing the Configuration Management scenario. GSG responded by developing a general multi-level approach to development of the CM scenario [61], and by proposing a specific tool-based approach for implementing the scenario [62].

The future requires further refinement and development of the approach for the Configuration Management scenario which will be achieved by:

- working with participating AFLC's to identify requirements
- Preparing a suitable NSW-based CM capability
- Testing the NSW-based CM capability prior to demonstration, and
- Supporting AFLC use of the CM capability during the demonstration period.

## 9.0 Notes and References

- [1] "NSW Tool Integration Studies: Final Report", GSG, January 5, 1978, Chapter 5
- [2] "NSW Management Plan", RADC, February 1, 1979
- [3] "Final Report on the National Software Works for the period October 1, 1977 through December 6, 1978", COMPASS, pp. 9-12
- [4] Ibid, p. 11
- [5] Op. Cit. [2]
- [6] "Policy Statement" net message, R. Metzger, November 24, 1979, pp. 1-2
- [7] Ibid., p. 2
- [8] Ibid., p. 2
- [9] "NSW Support Strategy" net message, R. Metzger, December 26, 1979
- [10] Op. Cit. [2], p. 1
- [11] Ibid., p. 1
- [12] Ibid., p. 2
- [13] Ibid., p. 7
- [14] Ibid., p. 5
- [15] Ibid., p. 5
- [16] Ibid., p. 14
- [17] "ARPANET Directory", Network Information Center, SRI, November 1980, pp. 9-10
- [18] Ibid., inside of back cover
- [19] Ibid., inside of front cover
- [20] "NSW 5.0 Test Plan", GSG, August 14, 1980

- [21] "NSW 5.0 Test Summary", GSG, September 11, 1980
- [22] "Methodology and Plan for Testing File Transformations", GSG, January 30, 1981
- [23] "Interactive Tool Testing Methodology", GSG, November 14, 1980
- [24] "Summary of Interactive Tool Tests: NSW Release 5.0", GSG, November 14, 1980
- [25] Op. Cit. [2]
- [26] "NSW Software Management and Control Plan", GSG, November 28, 1980
- [27] "NSW Users' Reference Manual System Release 5", COMPASS, August 1, 1980
- [28] "Interim NSW Managers' Tools", COMPASS, September 29, 1976
- [29] "NSW Tool Builder's Guide", COMPASS and BBN, February 18, 1977
- [30] "Tool Quality Management and Control Plan", IITRI, February, 1980
- [31] "Tool Installation Guide, Part 2", IITRI, May, 1980
- [32] "NSW User Impact Bulletin for System Release 5.0", GSG, March 25, 1981, pp. 73-74
- [33] Ibid., pp. 75-77
- [34] Op. Cit. [27]
- [35] "The NSW Users' Guide", COMPASS, December 30, 1976
- [36] Op. Cit. [28]
- [37] "NSW Beginners' Guide", GSG, August 1, 1978
- [38] Op. Cit. [32], pp
- [39] "NSW 4.1 User Impact Bulletin", GSG, January 31, 1980
- [40] Op. Cit. [32]

- [41] "UIB Entries for Inclusion in Revised User Reference Manual", GSG, February 14, 1980
- [42] "Updates to the NSW User Impact Bulletin for System Release 4.1", GSG, May 1, 1980
- [43] "NSW User Impact Bulletin Maintenance Plan", GSG, April 2, 1980
- [44] "Review of User Documentation for System Release 4.0", GSG, September 19, 1979
- [45] "Proposal to Upgrade NSW User Documentation", GSG, May 30, 1980
- [46] "Using the UCLA Native Language Processing Tool Kits", UCLA, July, 1979
- [47] "Using the DISPLAY Tool", UCLA, March, 1980
- [48] "Using the TSOEDIT Tool", UCLA, March, 1980
- [49] "Using the UCLA PL/I Tool Kit", UCLA, July, 1979
- [50] "Using the UCLA Interim Library Management Tool Kit", UCLA, July, 1979
- [51] "NSW Concept Paper", NSW Analysis Group (BBN and COMPASS), November 19, 1979
- [52] "NSW Requirements: Initial Ideas and Organization", GSG,
- [53] "A Review of the Conceptual NSW Design", GSG,
- [54] "NSW Functional Specification", NSW Analysis Group (BBN and COMPASS),
- [55] "NSW Operations Guide: System Release 4.1", GSG, October 31, 1980
- [56] "NSW HELP Facility System Specification", GSG, February 27, 1981
- [57] "NSW Documentation [Repository] Maintenance Manual", GSG, November 28, 1980
- [58] "NSW Technology Transfer Plan", GSG, February, 1980
- [59] "AFPLC Technology Demonstration Milestone Plan", GSG, February, 1980

- [60] "A Scenario Development Procedure for the AFLC Technology Demonstration, GSG, April, 1980
- [61] "An NSW-Based Configuration Management Scenario for the AFLC Technology Demonstration", GSG, June 9, 1980
- [62] "An NSW-Based Configuration Management Support Environment Using GIM/1100", GSG, June 9, 1980

## Appendix A: GSG Bibliography

Listed below (by author) are the major documents and reports prepared by GSG during the contract period covered by this report.

Bamberger, Frank, "Methodology and Plan for Testing NSW File Transformations", GSG, January 30, 1981

Bamberger, Frank, "NSW Requirements: Initial Ideas and Organization", GSG,

Bamberger, Frank, "NSW Global File Types and Transformations", GSG, July 10, 1980

Dingman, John, "Interactive Tool Testing Methodology", GSG, November 14, 1980

Dingman, J. and Gale, R., "NSW Operations Guide: System Release 4.1", GSG, October 31, 1980

Dingman, J. and Kneiss, P., "Summary of Interactive Tool Tests: NSW Release 5.0", GSG, November 14, 1980

Gale, Ross, "NSW 5.0 Test Plan", GSG, August 14, 1980

Gale, Ross, "NSW Documentation Summary", GSG, November 28, 1980

Gale, Ross, "A Scenario Development Procedure for the AFLC Technology Demonstration", GSG, April 1980

Gale, Ross, "A Review of the Release-Specific Documentation: System Release 4.1", GSG, January 1980

Gale, Ross, "A Review of the MSG User Manual", GSG, January 1980

Gale Ross, "NSW System Release and Assessment Procedure: Draft Proposal", GSG, January 1980

Gale, Ross, "NSW 4.1 System Status and Test Summary", GSG, February 12, 1980

Gale Ross, "PDC Comments", in "The NSW Application Experiment: Final Report", TRW,

Gale, R., Kneiss, P., Bamberger, F. and Dingman, J., "NSW 5.0 Test Summary", GSG, September 11, 1980

Gale, R., Payne, D., and Vincent, D., "An NSW-Based Configuration Management Scenario for the AFLC Technology Demonstration", GSG, June 9, 1980

Gale, R., Payne, D., and Vincent, D., "An NSW-Based Configuration Management Support Environment Using GIM/1100", GSG, June 9, 1980

Gale, R. and Bamberger, F., "NSW Fault Logger Design Review", GSG, March 1979

Gale, R. and Bamberger, F., "UNIX NSW Front End Design Review", GSG, February 1979

Gale, R. and Bamberger F., "A Review of the Conceptual NSW Design", GSG

Gale, R., Kneiss, P., and Bamberger, F., "NSW User Impact Bulletin for System Release 5.0", GSG, March 25, 1981

Kneiss, Peter, "NSW Tools On-line Documentation", GSG, April 2, 1980

Kneiss, Peter, "NSW User Impact Bulletin Maintenance Plan", GSG, March 19, 1980

Kneiss, Peter, "Review of User Documentation for System Release 4.0", GSG, September 19, 1979

Kneiss, Peter, "Proposal to Upgrade NSW User Documentation", GSG, May 30, 1980

Kneiss, Peter, "A Representative Tool Kit Using Networking Technology - Current NSW Tools", GSG

Kneiss, Peter, "UIB Entries for Inclusion in Revised User Reference Manual", GSG, February 14, 1980

Kneiss, P. and Bamberger, F., "NSW 4.1 User Impact Bulletin", GSG, January 31, 1980

Kneiss, P., Dingman, J., and Salisbury, S., "NSW Regression Tests for System Release 5.0", GSG, September 30, 1980

Kneiss, P., and Gale, R., "MONSTR Comments", GSG, June 1, 1980

Kneiss, P., and Dingman, J., "NSW Help Facility System Specification", GSG, February 27, 1981

Kneiss, P., and Bamberger, F., "Updates to the NSW User Impact Bulletin for System Release 4.1", GSG, May 1, 1980

Payne, Doug, "NSW Technology Transfer Plan", GSG, February, 1980

Payne, Doug, "AFLC Technology Demonstration Milestone Plan", GSG, February, 1980

Slocum, Lee, "NSW Software Management and Control Plan", GSG, April 14, 1981

"NSW Documentation [Repository] Maintenance Manual", GSG, November 28, 1980

## **Appendix B: Generic Configuration Items**

This appendix is designed to complement the discussion of NSw "Configuration Management" which appears in Section 5.6 of this report.

In the three (3) subsections which follow, this appendix summarizes the decomposition methodology used to identify "generic configuration items" and enumerates the current set of generic CI's for each of the following "packets":

- NSw System packet
- TOPS-20 Host Family packet. Due to the autonomous operational requirements of the IBM and Multics NSw hosts, the set of "packets" covered by this appendix excludes the Host Family packets for these host families.

### **B.1 Generic Configuration Items (CIs)**

The set of generic CIs for the NSW System consists of the following two (sets of) "packets":

- I. The "NSW System Packet", identifying the CIs generic to the (entire) NSW system, consists of the following three items:
  - (1) System databases list
  - (2) Operator aids list
  - (3) System documentation lists, one for each of the following sets of documentation:
    - (A) Host-independent functional specifications
    - (B) Release documentation
    - (C) User documentation
    - (D) Operations documentation
- II. [A set of] "Host Family Packets", identifying the CIs for each host family implementation of NSW functionality, consists of the following three (sets of) packets:
  1. A "Generic Host Packet", identifying the CIs common to one or more host components (see "Component Packets" below), consists of the following four items:
    - (1) Databases list
    - (2) Utility executables list
    - (3) Operator aids list
    - (4) Documentation list
  2. [A set of] "Component Packets", identifying the CIs for each host-specific component of NSW functionality, consists of the following five items:

- (1) Component executables list
  - (2) Component databases list
  - (3) Utility executables list
  - (4) Operator aids list
  - (5) Documentation list
3. [A set of] "Tool Packets", identifying the CIs for each NSW tool of the host family, consists of the following four items:
- (1) Tool executables list
  - (2) Tool databases list
  - (3) Operator aids list
  - (4) Documentation list

The generic CIs for the "NSW System" and the TOPS-20 Host Family "packets" are enumerated in subsections B.2 and B.3 which follow.

## B.2 NSW System Packet

The NSW system packet contains the following (sets of) generic configuration items:

- (1) System Database(s): CONFIG.BAS (A) [Universal Configuration Database (UCD)]
- (2) Operator Aid(s): none. ['none' = no configuration item(s) to be CONTROLLED]
- (3) System Documentation:
  - (A) Functional specifications
    - (1) "NSW Functional Specification" (A) ['A' = an up-to-date revision to be supplied by the Architecture Control Contractor (ACC) during release transition to the Product Development Contractor (PDC)].
    - (2) "System/Subsystem Specification" for each of the following generic functional components:
      - Batch Job Package (A)
      - Checkpointer (A)
      - Dispatcher (A)
      - Front End (A)
      - Fault Logger (A)
      - Foreman (A)
      - File Package (A)
      - MSG (A)
      - Operator Utility (A)
      - Works Manager (A)

- Works Manager Operator (A)
- (3) "Universal Configuration Database (UCD) Specification" (A)
- (B) Release Documentation
  - (1) "Release-Specific Documentation" [covering all host (family) implementations] (A)
- (C) User Documentation
  - (1) "NSw Beginners' Guide" (P+) ['P' = an up-to-date revision to be supplied by PDC; '+' = an up-to-date revision to be supplied shortly after release transition from PDC to the NSw user community].
  - (2) "NSw Users' Reference Manual" [revisions for both the TENEX/TOPS-20 and UNIX Front Ends] (A)
  - (3) "NSw Users' Guide" (A)
  - (4) "Interim NSw Manager's Tools" [until incorporated in the "NSw User's Reference Manual"] (A)
  - (5) "User Impact Bulletin" (O+) ['O' = up-to-date revision to be supplied by NSw Operations (NSWOPS)]
- (D) Operations Documentation
  - (1) "NSw Installation Guide" (A)
  - (2) "NSw Operations Guide" (O+)

### B.3 TOPS-20 Host Family Packet

The TOPS-20 Host Family packet includes the following (sets of) generic CI packets:

#### 1. TENEX/TOPS-20 Generic Host Packet:

- (1) Database(s): none
- (2) Utility Executable(s): LOGUTL.EXE (A)
- (3) Operator Aid(s): none
- (4) Documentation: none

#### 2. TENEX/TOPS-20 Component Packet(s):

##### Checkpoint:

- (1) Component Executable(s): CHKPTR.EXE (A)
- (2) Database(s): none
- (3) Utilities: none
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

##### Dispatcher:

- (1) Component Executable(s): DSPCHR.EXE (A) and NSWROOT.EXE (A)
- (2) Database(s): none
- (3) Utilities: none

- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

Front End:

- (1) Component Executable(s): FE.EXE (A), FETHDL.EXE (A) and UNTLNT.EXE (A)
- (2) Database(s): none
- (3) Utilities: none
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

Fault Logger

- (1) Component Executable(s): FL.EXE (A)
- (2) Database(s): none
- (3) Utilities: FLTEST.EXE (A) and FLOPER.EXE (A)
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

#### File Package

- (1) Component Executable(s): FLPKG.EXE (A)
- (2) Database(s): none
- (3) Utilities: none
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

#### Foreman

- (1) Component Executable(s): FOREMAN.EXE (A)
- (2) Database(s): none
- (3) Utilities: MKCOM.EXE (A), LOGRED.EXE (A)
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

#### MSG

- (1) Component Executable(s): MSG.EXE (A)
- (2) Database(s): MSG-GENERIC-NAMES (A),  
MSG-NETWORK-CONFIGURATION (A)
- (3) Utilities: none
- (4) Operator Aid(s): none

**(5) Documentation:**

- Interface Conformance Report(s): (A)
- Program Maintenance Manual(s): (A+)
- Operator Manual(s): (A)

**Operator Utility:**

- (1) Component Executable(s): OPRUTL.EXE (A)
- (2) Database(s): none
- (3) Utilities: none
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

**Works Manager:**

- (1) Component Executable(s): WM.EXE (A)
- (2) Database(s): WM-ONLINE.TABLES [skeleton] (A), DBWTF.1 [skeleton] (A), DBFCE.1 [skeleton] (A)
- (3) Utilities: SIMWMT.EXE (A), SIMWTF.EXE (A), SIMINF.EXE (A), DMPUTL.EXE (A), DB3TAT.EXE (A), TBLTRN.EXE (A) [Note: Database conversion procedures should be included in the "NSW Installation Guide" (see "System Documentation Packet")]
- (4) Operator Aid(s): DO files for creating:
  - The Checkpointer Control Entry in the Works Manager Database
  - Database skeletons for WM-ONLINE.TABLES, DBWTF.1 and DBFCE.1

(5) Documentation:

- Interface Conformance Report(s): (A)
- Program Maintenance Manual(s): (A+)
- Operator Manual(s): (A)

works Manager Operator:

- (1) Component Executable(s): WMO.EXE (A)
- (2) Database(s): none
- (3) Utilities: WMOUtl.EXE (A)
- (4) Operator Aid(s): none
- (5) Documentation:
  - Interface Conformance Report(s): (A)
  - Program Maintenance Manual(s): (A+)
  - Operator Manual(s): (A)

3. TENEX/TOPS-20 Tool Packet(s):

BCPL:

- (1) Tool Executable(s): BCPL.EXE (T) ['T' = an up-to-date revision to be supplied by the NSW Tool Manager]
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified

- Native Host: to be identified

**BDDT:**

- (1) Tool Executable(s): BDDT.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**CONCORDANCE:**

- (1) Tool Executable(s): CONCORDANCE.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**DESCRIBE:**

- (1) Tool Executable(s): DESCRIBE.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database

(4) Documentation:

- Installation: to be identified
- NSW-Specific: to be identified
- Native Host: to be identified

ECL:

(1) Tool Executable(s): ECL.EXE (T)

(2) Database(s): to be identified

(3) Operator Aid(s): DO files for creating tool  
and global file descriptor entries in the  
works Manager database

(4) Documentation:

- Installation: to be identified
- NSW-Specific: to be identified
- Native Host: to be identified

EMLOAD:

(1) Tool Executable(s): EMLoad.EXE (T)

(2) Database(s): to be identified

(3) Operator Aid(s): DO files for creating tool  
and global file descriptor entries in the  
works Manager database

(4) Documentation:

- Installation: to be identified
- NSW-Specific: to be identified
- Native Host: to be identified

FMIBCPL:

- (1) Tool Executable(s): FMBCPL.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

FTP:

- (1) Tool Executable(s): FTP.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

HOSTAT:

- (1) Tool Executable(s): HOSTAT.EXE (T),  
NETSTAT.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:

- Installation: to be identified
- NSW-Specific: to be identified
- Native Host: to be identified

IDDT:

- (1) Tool Executable(s): IDDT.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

JIGSAW:

- (1) Tool Executable(s): JIGSAW.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

LINKER:

- (1) Tool Executable(s): LINKER.EXE (T)

- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

MACRO:

- (1) Tool Executable(s): MACRO.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

MACRO20:

- (1) Tool Executable(s): MACRO20.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified

- Native Host: to be identified

**MRUNOFF:**

- (1) Tool Executable(s): MRUNOFF.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**NETSTAT:**

- (1) Tool Executable(s): NETSTAT.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**PRIM:**

- (1) Tool Executable(s): PRIM.EXE (T), IO-SERVER-FOR-PRIM.EXE (T), GPM.EXE (T), UDDT.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database

(4) Documentation:

- Installation: to be identified
- NSW-Specific: to be identified
- Native Host: to be identified

PSAVE:

- (1) Tool Executable(s): PSAVE.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

SOS:

- (1) Tool Executable(s): SOS.EXE (T), SOSHLP.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**SPELL:**

- (1) Tool Executable(s): SPELL.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**SRCCOM:**

- (1) Tool Executable(s): SRCCOM.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**TECO:**

- (1) Tool Executable(s): TECO.EXEC (T), EXEC.EXE (T)
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the Works Manager database
- (4) Documentation:

- Installation: to be identified
- NSW-Specific: to be identified
- Native Host: to be identified

(2) Database(s): to be identified

**UYK20:**

- (1) Tool Executable(s): UYK20.EXE (T),  
IO-SERVER-FOR-PRIM.EXE (T), GPM.EXE (T),  
UDDT.EXE (T), PRIM.EXE (T)
- (3) Operator Aid(s): DO files for creating tool  
and global file descriptor entries in the  
Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**U1050:**

- (1) Tool Executable(s): U1050.EXE (T),  
IO-SERVER-FOR-PRIM.EXE (T), GPM.EXE (T),  
UDDT.EXE (T), PRIM.EXE (T)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool  
and global file descriptor entries in the  
Works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSW-Specific: to be identified
  - Native Host: to be identified

**XED:**

- (1) Tool Executable(s): XED.EXE(I)
- (2) Database(s): to be identified
- (3) Operator Aid(s): DO files for creating tool and global file descriptor entries in the works Manager database
- (4) Documentation:
  - Installation: to be identified
  - NSw-Specific: to be identified
  - Native Host: to be identified

## Appendix C: Overview of the NSW HELP Facility

[Note: The overview of goals, requirements and capabilities of the NSW Help Facility (including execution examples) which follows was extracted from "The NSW HELP Facility System Specification" by Peter Kneiss and John Dingman, dated February 27, 1981.]

### NSW Help Facility

The NSW system consists of many dissimilar hosts and internal functions supported by a variety of development contractors. Each tool purveyor, host operating system developer, and each NSW contractor documents a particular concern independent of its integration in NSW. As a result, the documentation of the conglomerate of systems that is seen by the user as one entity originates as a series of separate publications. Because the documentation for various aspects of NSW has many origins, documentation standards cannot be imposed to accommodate the HELP facility. Rather, the HELP facility must function as a delivery facility that would consolidate access to the existing documentation. Thus, the following broad goals were set when specifying the information delivery system for the NSW HELP facility:

- To standardize the user interface to conform with other NSW components

- To centralize information in one source instead of many tools for many types of information

- To create a generalized interface which is adaptable to a variety of information files

- To create a facility whose files and topic selection structure can be externally configured by an NSW Operator (i.e., to have few, if any parameters, topics, etc., compiled into the software itself).

In addition, there were two functions which the HELP facility needed to provide to users:

Enable a user logged into NSW to send a message to an NSW operator (who is not necessarily logged in). This would allow HELP users to document bugs, make requests, or ask procedural questions without leaving NSW. In particular, these messages could be entered during the times an operator is not available by phone.

Display the status of NSW hosts so users can be informed when remote hosts are not responding, thus avoiding having to wait for connection timeouts, and providing affirmation that long running tasks are actually continuing to execute.

The HELP facility currently installed in the User System satisfies the above mentioned goals. It aids users in selecting the type of information needed from among a variety of information files. At the top level of the interface, the user may also choose to send a message to Ops (TELOPS) or display the most recent status of NSW hosts (STATUS). The tool is designed for flexibility in changing the topical map (e.g., to reflect changes in NSW) and for adaptability in displaying different information files. The current version can provide a single online access point to the information previously provided by the USER IMPACT BULLETIN, the NSW DOCUMENTATION SUMMARY, the NSW USER REFERENCE MANUAL, the NSW tools DESCRIBE and HELP-RM, the RADC-20 Command INFORMATION (about) NSW-STATUS, and the INTERIM NSW MANAGER'S TOOLS.

The following typescript illustrates some of the features of the HELP facility. A dollar-sign (\$) indicates that an <escape> was entered. The user's lower case typein can usually be distinguished from HELP's uppercase retyping of the command line. It should be noted that all of the menus are functions of the database rather than the program and can be changed by altering the help files. The only compiled in commands are "TELOPS", "STATUS", and the display commands prompted for with "(H,P,S,E, or ?)=>".

} Our comments are offset in this manner

```
} The following typescript hypothesizes
} that a naive user tries to use a tool
} and encounters a problem. He attempts to "quit"
} NSW and encounters one of the few times that the
} TOPS-20 Front End does not help him with command
} syntax. Instead of the help NSW usually
} provides in response to an escape, he
} receives an error message.
```

) we show how the HELP facility would lead him out  
) of his quandary if the Front End response  
) included a pointer to the HELP tool.

NSW User System Version 5.0  
Report problems to NSWOPSERADC-20, (315)330-6256.

NSW: use qedx-rms (Confirm): !

>>> Tue 17-Feb-81 08:10:47-PST: No rights for tool QEDX-RM  
>>> Tue 17 Feb 1981 0810:49-PST -- (CA.FE(WMTALK))  
>> QEDX-RM not available

NSW: quits NO TOOLS EXIST

NSW: ?

Options are:

COPY  
DELETE  
LOGOUT  
NET  
USE  
SEMAPHORE  
RENAME  
JOB  
PASSWORD  
FASTOUT  
SHOW  
ALTER  
MOVELOG  
QUIT  
RESUME

[For more information, type "use help"]  
continue...!

NSW:

) Taking his cue from the NSW Front End's  
) response to a question mark, the user invokes  
) the HELP tool and types "?"  
) for a list of HELP topics.

NSW: use helps (Confirm): !

[Connecting to 2HELP...]

NSW HELP FACILITY

Type ?<CR> for list of topics

HELP>?

Topics/commands available are:

COMMANDS -	describes and shows examples of NSW commands.
NEWS -	allows users to read notices and news entered by NSWOPS concerning schedules, new tools, etc.
POINTERS -	displays NSW files which users may access for additional information.
QUIT -	exits tool and returns to NSW command level.
STATUS -	displays current status of NSW hosts.
TELOPS -	a device which enables NSW users to send questions, complaints, or comments to NSWOPS.
TOOLS -	describes NSW tools, outstanding problems, and tool rights.

Only enough of a command to make it unique must be entered.

Entering "?" will show possible commands/subcommands.

Fields (between commands and subcommands) must be separated by a space.

For example:

"c ?<CR>" will list the possible subcommands of 'COMMANDS'.

"to ?<CR>" will list the possible subcommands of 'TOOLS'.

Use the following characters to edit your input:

<DEL> - delete last character

CTRL-W - delete last word

CTRL-U - delete line

CTRL-R - retype line

CTRL-V - quote next character

} It might occur to a user who has been warned  
} that NSW is a multi-host system, that the host for  
} his tool is down. The HELP facility's  
} display of host status eliminates that possibility.

HELP>status

NSw User System Status as of: 13-Feb-81 1711-EST

USC-ISLE (works Manager Host) is UP  
USC-ISIC (Tool Bearing Host) is UP  
RADC-MULTICS (Tool Bearing Host) is UP  
UCLA-CCN (Tool Bearing Host) will be down until  
further notice due to system upgrading.

} If it occurs to the user to check the command  
} "QUIT" to see if it is intended to end NSw  
} sessions, the following might ensue. (The  
} typescript illustrates HELP's interpretation  
} of escape and question mark.)

HELP>commandss

COMMAND NAME (13 CHOICES)

HELP>COMMANDS quits

QUIT (NSw COMMAND) (4 CHOICES)

HELP>COMMANDS QUIT ?

QUIT (NSw COMMAND) (4 CHOICES)

DESCRIPTION (OF COMMAND)

EXAMPLE (OF COMMAND USES)

NOTES

EXCEPTIONS

HELP>COMMANDS QUIT description

COMMAND: QUIT (NSw Command) (10 lines) (NURM 5.0)

PURPOSE: End a pending tool instance.

SYNTAX:

user	system	user	system	user
----	-----	----	-----	----
QUIT	toolname	<name>	action	{ ** SUSPEND { ABORT { TERMINATE

[End of items.]

- > The user, having seen the four topics
- > under the NSW QUIT command, now
- > can use a minimum number of characters to
- > specify displaying "COMMAND QUIT NOTES".

HELP>c q n

FUNCTIONAL NOTES: QUIT (NSW Command ) (27 lines) (NURM 5.0)

1. For <name> you should type the tool instance name. Typing "?" at this point will yield a list of pending tool instances.

2. As explained in the NSW Users' Manual:

[\*\* SUSPEND causes the tool to cease activity; if the tool permits, it can be reactivated (see RESUME) -- Not yet implemented. \*\*]

ABORT kills the tool instance immediately; in particular, files which have been created may not be delivered into NSW filespace.

TERMINATE causes appropriate tools to enter their termination code; whether or not files are delivered depends on the characteristics of the tool.

3. If SUSPEND or TERMINATE is not meaningful for a particular tool the user is so notified.

[End of items.]

- > If an information file contains multiple
- > topics (as indicated by the presence of form
- > feeds), HELP offers the four display options
- > illustrated below.

HELP>c q exceptions

QUIT (NSW Command)

(3 items)

[UIB 5.0]

(H,P,S,E, or ?)=>?

- |                      |                                |
|----------------------|--------------------------------|
| H                    | - Show headers of all items    |
| P<CR> (or just <CR>) | - Print this item              |
| Pn<CR>               | - Print item number n          |
| S                    | - Skip to next item            |
| E (or Ctrl-X)        | - Return to HELP command level |
| ?                    | - Print this text              |

(H,P,S,E, or ?)=>h

- 1 - Never use the QUIT command on a CCN interactive tool
- 2 - It is inadvisable to slew-away from a tool that has begun a termination sequence (i.e., the tool's termination command has been entered).
- 3 - The following sequence of messages may be generated when trying to QUIT or RESUME a tool-instance:

[End of items.]

(H,P,S,E, or ?)=>p1

Never use the QUIT command on a CCN interactive tool, (e.g., DISPLAY-UC). Instead you should use the tool's termination command (e.g., "END"). Frequently as a consequence of using QUIT on a CCN interactive tool, your Front End is damaged and NSW is unable to clear the CCN system resources for that tool-instance. Since these resources are finite, tool-instances which QUIT fails to clean-up will accumulate until they exhaust the available NSW resources at CCN.  
[NST-626] & [NST-592]

-----o-----

It is inadvisable to slew-away from a tool that has begun a termination sequence (i.e., the tool's termination command has been entered).

(H,P,S,E, or ?)=>s

The following sequence of messages may be generated when trying to QUIT or RESUME a tool-instance:

(H,P,S,E, or ?)=>e

```

} If the user went on to retrieve information about
} tool rights, or looked up the error message in
} an error message dictionary, he should find
} that his real problem is obtaining rights for
} the tool. To do this he needs to communicate
} with the NSW operator.

```

HELP>telops

Type Control-X to abort, "?<CR>" for help.

Please enter your name

=>Peter Kneiss

Enter your Project.Node name

=>gsq.admin

Enter your net address (<CR> if none)

=>kneiss@radc-20

Enter subject (one line summary of problem or topic)

=>Grant rights to QEDX-RM

Enter your message below on as many lines as needed, terminating  
your input with Control-Z or ESCAPE.

Please grant GSG.ADMIN the rights to QEDX-RM.

Thanks

^Z

Your message is being delivered to NSWOPS. A reply will be sent to  
you soon. Thank you.

HELP>quits [Confirm QUIT]:

NSW:

# Appendix D: Sample "Weekly Usage Report"

## WEEKLY USAGE REPORT FOR THE NSW USER SYSTEM (VERSION 5.0)

From MON MAR 09 1981 0000EST to MON MAR 16 1981 0000EST.

### I. Number of Logins:

MON	TUE	WED	THU	FRI	SAT	SUN	*TOTAL*
4	10	2	7	7	0	0	30

### II. Node Usage Summary:

PROJECT + NODE  
TOOLS USED

```

-----
ACC + HIS      17 Logins Totaling 04:53:00
                HELP: 2 Sessions Using 00:00:06 in 00:44:00
                PL1-RM: 2 Sessions Using 00:00:04 in 00:12:00
                QEDX-RM: 3 Sessions Using 00:00:01 in 00:01:00
                RUN-RM: 9 Sessions Using 00:00:36 in 00:48:00
-----
AFLC + CRAIG   3 Logins Totaling 01:36:00
                HELP-RM: 1 Sessions Using 00:00:04 in 00:12:00
-----
GSG + ADMIN    3 Logins Totaling 05:42:00
                DESCRIBE-IE: 1 Sessions Using 00:00:01 in 00:02:00
                FORTRAN-RM: 1 Sessions Using 00:00:00 in 00:00:00
                FTP-IE: 1 Sessions Using 00:00:03 in 00:13:00
                HELP: 1 Sessions Using 00:00:01 in 00:02:00
                XED-IE: 1 Sessions Using 00:00:03 in 00:12:00
-----
GSG + JRD      3 Logins Totaling 00:22:00
                HELP: 6 Sessions Using 00:00:01 in 00:04:00
-----
GSG + KNEISS   9 Logins Totaling 02:54:00
                SOS-R2: 3 Sessions Using 00:00:54 in 00:42:00
                XED-IC: 1 Sessions Using 00:00:00 in 00:01:00
                XED-IE: 2 Sessions Using 00:00:00 in 00:00:00
                XED-R2: 6 Sessions Using 00:00:04 in 00:36:00
-----

```

```

RADC + ADMIN      1 Logins Totaling 00:27:00
  No tool usage
-----
RADC + BASKINGER  1 Logins Totaling 00:56:00
  HELP:           1 Sessions Using 00:00:01 in 00:31:00
  RUN-RM:         1 Sessions Using 00:00:00 in 00:00:00
-----

```

### III. Tool Usage Summary:

TOOLNAME	TOTAL SESSIONS	TOTAL CPU	TOTAL CONNECT
-----	-----	-----	-----
DESCRIBE-IE	1	00:00:01	00:02:00
FORTRAN-RM	1	00:00:00	00:00:00
FTP-IE	1	00:00:03	00:13:00
HELP	10	00:00:09	01:21:00
HELP-RM	1	00:00:04	00:12:00
PL1-RM	2	00:00:04	00:12:00
QEDX-RM	3	00:00:01	00:01:00
RUN-RM	10	00:00:36	00:48:00
SOS-R2	3	00:00:54	00:42:00
XED-IC	1	00:00:00	00:01:00
XED-IE	3	00:00:03	00:12:00
XED-R2	6	00:00:04	00:36:00

### IV. NSW Up-time/Down-time (wMH only, TBHs excluded)

The following conventions will be used in this section:

U-<TIME> -- Time NSW came up (all times are EST)  
D-<TIME> -- Time NSW went down  
Number in parentheses following time indicates reason NSW  
went down, using the following codes:

- (1) -- ISIE scheduled downtime
- (2) -- Unknown ISIE system failure
- (3) -- Known NSW failure
- (4) -- Unknown NSW failure
- (5) -- Other

A single asterisk printed under any day means the system was  
unavailable the entire day. A double asterisk means the system  
was up the entire day.

MON	TUE	WED	THU	FRI
---	---	---	---	---
U-1025	D-1130(2)	**	D-0527(2)	U-0836
D-1445(2)	U-1229		U-0900	D-1809(2)
U-1635			D-1459(2)	
			U-1510	

D-2106(2)

SAT  
---  
\*

SUN  
---  
\*

Steve Salisbury (NSWOPS@RADC-20)

addresses	number of copies	line number
Leon McDowell RADC/ISCP	6	
RADC/TSLD GRIFFISS AFB NY 13441	1	2
RADC/DAP GRIFFISS AFB NY 13441	2	3
ADMINISTRATOR DEF TECH INF CTR ATTN: DTIC-DDA CAMERON STA BG 5 ALEXANDRIA VA 22314	12	5
HQ ESC (XPZP) SAN ANTONIO TX 78243	1	12
HQ ESC/D00 SAN ANTONIO TX 78243	1	13
HQ USAF/XOKT WASHINGTON DC 20330	1	17
HQ USAF/RDST WASHINGTON DC 20330	1	20
HQ USAF/RDPV WASHINGTON DC 20330	1	21

PENTAGON  
USLR&E, RM 3D-139  
ATTN: TSCO  
WASHINGTON DC 20301

2 26

0 29

HQ AFSC/DLAE  
ANDREWS AFB DC 20334

1 30

HQ AFSC/XRK  
ANDREWS AFB DC 20334

1 37

0 38

HQ SAC/NRI (STINFO LIBRARY)  
OFFUTT AFB NE 68113

1 39

0 44

HQ 3246 TW/TETW  
EGLIN AFB FL 32542

1 46

AFATL/DLODL  
EGLIN AFB FL 32542

1 47

ESMC/PM (STINFO)  
PATRICK AFB FL 32925

1 48

TAFIG/IIPB LANGLEY AFB VA 23665	1	50
HQ TAC/XPS (STINFO) LANGLEY AFB VA 23665	1	51
HQ TAC/XPJC ATTN: Lt Taylor LANGLEY AFB VA 23665	1	52
TAFIG/IICJ ATTN: Capt John Morrison LANGLEY AFB VA 23665	2	53
	0	55
HQ TAC/DECG LANGLEY AFB VA 23665	1	56
HQ TAC/JRF LANGLEY AFB VA 23665	1	58
AFSC LIAISON OFFICE LANGLEY RESEARCH CENTER (NASA) LANGLEY AFB VA 23665	1	59
AFWL/NTYEE ( C E BAUM ) KIRTLAND AFB NM 87117	1	63
AFWL/SUL ATTN: TECHNICAL LIBRARY KIRTLAND AFB NM 87117	1	64

ASL/ENEGE 1 67  
ATTN: CAPT T CLELAND  
WRIGHT-PATTERSON AFB OH 45433

ASL/ENEGE 1 68  
ATTN: MR LARRY WEAVER  
WRIGHT-PATTERSON AFB OH 45433

ASD/XRE 1 80  
WRIGHT-PATTERSON AFB OH 45433

1  
AFIT/LDE - TECHNICAL LIBRARY 1 81  
BUILDING 640, AREA B  
WRIGHT-PATTERSON AFB OH 45433

AFHRL/OTN 1 90  
WILLIAMS AFB AZ 85224

AFHRL/OA 1 91  
BROOKS AFB TX 78235

AUL/ISE 67-342 1 96  
MAXWELL AFB AL 36112

HQ AFCC/DAPL 1 98  
BLDG P-40 NORTH, RM 9  
SCOTT AFB IL 62225

HQ AFCC/EPE 1 100  
SCOTT AFB IL 62225

AFHRL/LRT LOWRY AFB CO 80230	1	101
3300 TTW/TTGX KEESLER AFB MS 39534	1	103
DEFENSE INTELLIGENCE AGENCY ATTN: RSE-2 (LT COL SCHWARTZ) WASHINGTON DC 20301	1	107
DEFENSE INTELLIGENCE AGENCY ATTN: RSM-1 WASHINGTON DC 20301	1	108
CODE R123 TECHNICAL LIBRARY DEFENSE COMMUNICATIONS ENGINEERING CENTER 1860 WIEHLE AVENUE RESTON VA 22090	1	110
DIRECTOR DEFENSE NUCLEAR AGENCY ATTN: TIL WASHINGTON DC 20305	1	111
CHIEF, C3 DIVISION DEVELOPMENT CENTER, MCDEC ATTN: R S HARTMAN QUANTICA VA 22134	2	112
AFLMC/LGY ATTN: MAJOR MORGAN GUNTER AFS AL 36114	1	116
DIRECTOR BMD ADVANCED TECHNOLOGY CENTER ATTN: ATC-P, CHARLES VICK PO BOX 1500 HUNTSVILLE AL 35807	1	119

BOARD/CMJ TECHNICAL LIBRARY FL 2878 BOX 14 FPO NY 09510	21	120
COMMANDING OFFICER NAVAL AVIONICS CENTER LIBRARY - CODE 765 INDIANAPOLIS IN 46218	1	123
NAVAL TRAINING EQUIPMENT CENTER TECHNICAL INFORMATION CENTER ORLANDO FL 32813	1	124
COMMANDER NAVAL OCEAN SYSTEMS CENTER ATTN: TECHNICAL LIBRARY, CODE 4473B SAN DIEGO CA 92152	1	125
SUPERINTENDENT (CODE 1424) NAVAL POSTGRADUATE SCHOOL MONTEREY CA 93940	1	127
COMMANDING OFFICER NAVAL RESEARCH LABORATORY CODE 2627 WASHINGTON DC 20375	1	128
REDSTONE SCIENTIFIC INFORMATION CENTER ATTN: DRSMI-RPRD US ARMY MISSILE COMMAND REDSTONE ARSENAL AL 35809	2	131
DOT/FAA TECHNICAL CENTER ARL-142 (ATTN: A R CIOFFI) ATLANTIC CITY NJ 08405	1	134
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH MESA LIBRARY PO BOX 3000 BOULDER CO 80307	1	135

AIR FORCE ELEMENT (AFELM) THE RAND CORP 1700 MAIN STREET SANTA MONICA CA 90406	1	140
DR RAYNER K ROSICH ELECTRO MAGNETIC APPLNS, INC C/O 7031 PIERSON STREET ARVADE CO 80004	1	142
AELC LIBRARY (TECH FILES) ARNOLD AFS TN 37389	1	143
Director National Security Agency ATTN: T1213/TDL Fort Meade MD 20755	0	144
Director National Security Agency ATTN: W07 Fort Meade MD 20755	1	145
Director National Security Agency ATTN: W31 Fort Meade MD 20755	1	148
Director National Security Agency ATTN: R03 Fort Meade MD 20755	1	155
Director National Security Agency ATTN: R1 Fort Meade MD 20755	1	156
Director National Security ATTN: R2 Fort Meade MD 20755	1	157

Director National Security Agency ATTN: R5 Fort Meade MD 20755	1	158
Director National Security Agency ATTN: R6 Fort Meade MD 20755	1	159
Director National Security Agency ATTN: R7 Fort Meade MD 20755	1	160
Director National Security Agency ATTN: R8 Fort Meade MD 20755	1	161
Colonel Larry Druffel ARPA/ITTO 1400 Wilson Blvd Arlington VA 22209	1	63
HQ ESD/FAE, STOP 27 HANSCom AFB MA 01731	1	164
ESD/DCKD (STOP 53) ATTN: LT COMBS HANSCom AFB MA 01731	1	168
HQ ESD/YSEA HANSCom AFB MA 01731	1	172
ESD/XRET HANSCom AFB MA 01731	1	174

ESD/XRWS  
HANSCOM AFB MA 01731

1 178

ESD/XRWT  
HANSCOM AFB MA 01731

1 179

ESD/XRWW  
HANSCOM AFB MA 01731

1 180

ESD/XRTR  
HANSCOM AFB MA 01731

1 181

ESD/XR  
HANSCOM AFB MA 01731

1 184

ESD/DCR-3E  
HANSCOM AFB MA 01731

1 185

HQ ESD/YSM (STOP 18)  
HANSCOM AFB MA 01731

2 186

HQ ESD/DCR-1S  
HANSCOM AFB MA 01731

1 187

HQ ESD/DCR-1I  
HANSCOM AFB MA 01731

1 188

ESD/TOFF (Capt Cliff Gardner)  
HANS COM AFB MA 01731

2 3

AFIC/LOEC (Capt Bill Riski)  
WRIGHT-PATTERSON AFB OH 45433

2 4

WR-ALC/MMECDM (Palmer Craig)  
ROBINS AFB GA 31098

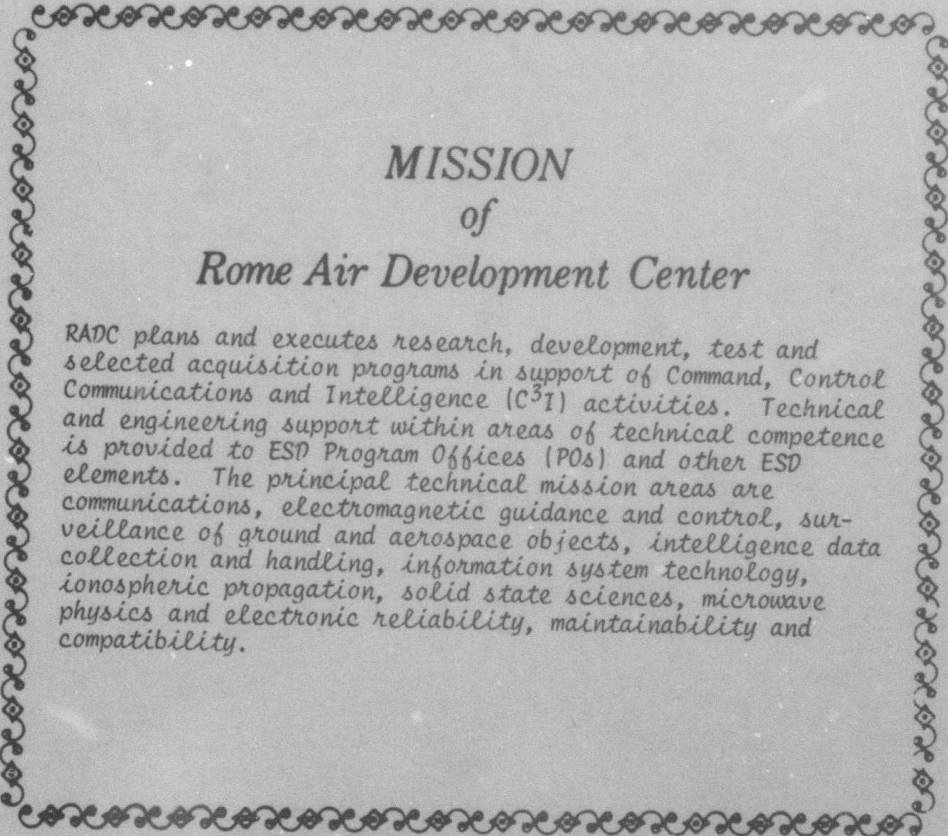
3 5

SM-ALC/MMECF (Van Johnson)  
McCLELLAN AFB CA 95652

3 6

OC-ALC/MMEC (Mike Parish)  
TINKER AFB OK 73145

3 7

A decorative border with a repeating scroll-like pattern surrounds the central text.

## *MISSION of Rome Air Development Center*

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C<sup>3</sup>I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.